

iWay

iWay XML Adapter for RDBMS User's Guide
Version 5 Release 5

Updated for J2EE CA 1.5

EDA, EDA/SQL, FIDEL, FOCCALC, FOCUS, FOCUS Fusion, FOCUS Vision, Hospital-Trac, Information Builders, the Information Builders logo, Parlay, PC/FOCUS, SmartMart, SmartMode, SNAPPack, TableTalk, WALDO, Web390, WebFOCUS and WorldMART are registered trademarks, and iWay and iWay Software are trademarks of Information Builders, Inc.

Sun and Java are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

Due to the nature of this material, this document refers to numerous hardware and software products by their trademarks. In most, if not all cases, these designations are claimed as trademarks or registered trademarks by their respective companies. It is not this publisher's intent to use any of these names generically. The reader is therefore cautioned to investigate all claimed trademark rights before using any of these names other than to refer to the product described.

Copyright © 2005, by Information Builders, Inc and iWay Software. All rights reserved. Patent Pending. This manual, or parts thereof, may not be reproduced in any form without the written permission of Information Builders, Inc.

Preface

This document is intended for system integrators who develop client-server interfaces between RDBMS and third-party enterprise information system (EIS) applications.

How This Manual Is Organized

The following table lists the numbers and titles of the chapters and appendix for this manual with a brief description of the contents of each chapter and appendix.

Chapter/Appendix		Contents
1	Introducing the iWay XML Adapter for RDBMS	Provides an overview of the adapter and how it works.
2	Creating XML Schemas or Business Services	Describes how to create schemas for RDBMS SQL statements and stored procedures for Web services or for JCA deployment.
3	Listening for Database Events	Describes how to configure a listener to listen to a database event.
4	Using Web Services Policy-Based Security	Describes how to configure Web services policy-based security.
5	Management and Monitoring	Describes how to configure and use monitoring tools provided by iBSE and JCA to gauge the performance of your run-time environment.
A	JDBC Drivers	Lists the supported JDBC drivers for use with the adapter.

Documentation Conventions

The following table lists and describes the conventions that apply throughout this manual.

Convention	Description
<code>THIS TYPEFACE</code> or <code>this typeface</code>	Denotes syntax that you must enter exactly as shown.
<code>this typeface</code>	Represents a placeholder (or variable) in syntax for a value that you or the system must supply.
<code>underscore</code>	Indicates a default setting.
<code>this typeface</code>	Represents a placeholder (or variable) in a text paragraph, a cross-reference, or an important term.
this typeface	Highlights a file name or command in a text paragraph that must be lowercase.
<code>this typeface</code>	Indicates a button, menu item, or dialog box option you can click or select.
Key + Key	Indicates keys that you must press simultaneously.
{ }	Indicates two or three choices; type one of them, not the braces.
	Separates mutually exclusive choices in syntax. Type one of them, not the symbol.
...	Indicates that you can enter a parameter multiple times. Type only the parameter, not the ellipsis points (...).
.	Indicates that there are (or could be) intervening or additional commands.

Related Publications

To view a current listing of our publications and to place an order, visit our World Wide Web site, <http://www.informationbuilders.com>. You can also contact the Publications Order Department at (800) 969-4636.

Customer Support

Do you have questions about the iWay XML Adapter for RDBMS?

If you bought the product from a vendor other than iWay Software, contact your distributor.

If you bought the product directly from iWay Software, call Information Builders Customer Support Service (CSS) at (800) 736-6130 or (212) 736-6130. Customer Support Consultants are available Monday through Friday between 8:00 a.m. and 8:00 p.m. EST to address all your iWay XML Adapter for RDBMS questions. Information Builders consultants can also give you general guidance regarding product capabilities and documentation. Please be ready to provide your six-digit site code number (xxxx.xx) when you call.

You can also access support services electronically, 24 hours a day, with InfoResponse Online. InfoResponse Online is accessible through our World Wide Web site, <http://www.informationbuilders.com>. It connects you to the tracking system and known-problem database at the Information Builders support center. Registered users can open, update, and view the status of cases in the tracking system and read descriptions of reported software issues. New users can register immediately for this service. The technical support section of www.informationbuilders.com also provides usage techniques, diagnostic tips, and answers to frequently asked questions.

To learn about the full range of available support services, ask your Information Builders representative about InfoResponse Online, or call (800) 969-INFO.

Help Us to Serve You Better

To help our consultants answer your questions effectively, please be prepared to provide specifications and sample files and to answer questions about errors and problems.

The following tables list the specifications our consultants require.

Platform	
Operating System	
OS Version	
Product List	
Adapters	
Adapter Deployment	For example, JCA, Business Services Engine, iWay Adapter Manager
Container Version	

The following table lists components. Specify the version in the column provided.

Component	Version
iWay Adapter	
EIS (DBMS/APP)	
HOTFIX / Service Pack	

The following table lists the types of Application Explorer. Specify the version (and platform, if different than listed previously) in the columns provided.

Application Explorer Type	Version	Platform
Swing		
Servlet		
ASP		

In the following table, specify the JVM version and vendor in the columns provided.

Version	Vendor

The following table lists additional questions to help us serve you better.

Request/Question	Error/Problem Details or Information
Provide usage scenarios or summarize the application that produces the problem.	
Did this happen previously?	
Can you reproduce this problem consistently?	
Any change in the application environment: software configuration, EIS/ database configuration, application, and so forth?	

Request/Question	Error/Problem Details or Information
Under what circumstance does the problem <i>not</i> occur?	
Describe the steps to reproduce the problem.	
Describe the problem .	
Specify the error message(s).	

The following table lists error/problem files that might be applicable.

XML schema
XML instances
Other input documents (transformation)
Error screen shots
Error output files
Trace and log files
Log transaction

User Feedback

In an effort to produce effective documentation, the Documentation Services staff welcomes your opinions regarding this manual. Please use the Reader Comments form at the end of this manual to communicate suggestions for improving this publication or to alert us to corrections. You also can go to our Web site, <http://www.iwaysoftware.com> and use the Documentation Feedback form.

Thank you, in advance, for your comments.

iWay Software Training and Professional Services

Interested in training? Our Education Department offers a wide variety of training courses for iWay Software and other Information Builders products.

For information on course descriptions, locations, and dates, or to register for classes, visit our World Wide Web site, <http://www.iwaysoftware.com> or call (800) 969-INFO to speak to an Education Representative.

Interested in technical assistance for your implementation? Our Professional Services department provides expert design, systems architecture, implementation, and project management services for all your business integration projects. For information, visit our World Wide Web site, <http://www.iwaysoftware.com>.

Contents

1. Introducing the iWay XML Adapter for RDBMS	1-1
Introduction	1-2
Using the Adapter With Relational Databases	1-2
Using the Adapter to Send a Request	1-3
Using the Adapter Listener	1-3
Using the Adapter to Access Non-relational Databases	1-5
Accessing Stored Procedures for Non-relational Data Sources	1-7
Deployment Information for the Adapter	1-7
Deployment Information Roadmap	1-8
The iWay Business Services Engine (iBSE)	1-8
The iWay Enterprise Connector for J2EE Connector Architecture (JCA)	1-9
2. Creating XML Schemas or Business Services	2-1
Generating Schemas and a Business Service	2-2
Starting Servlet Application Explorer	2-3
Creating and Managing a Connection	2-4
Disconnecting From a Defined Target	2-9
Editing a Defined Target	2-10
Deleting a Defined Target	2-11
Viewing Metadata	2-11
Creating a Statement and Generating Schemas	2-14
Creating and Testing a Parameterized SQL Statement	2-17
Creating a Batch Statement	2-21
Generating a Schema for a Prepared Statement	2-22
Generating a Schema for a Stored Procedure	2-23
Request and Response Documents	2-24
Regular SQL Statements	2-24
Parameterized SQL Statements	2-27
Stored Procedure Schemas for an Oracle Database	2-30
Stored Procedure Schemas for a VSAM Database	2-31
Multiple Executions of the Same SQL Statement or Stored Procedure	2-34
Stored Procedures with Constraints	2-36
Understanding iWay Business Services	2-41
Creating a Business Service	2-41
Testing a Business Service	2-43
Generating WSDL From a Web Service	2-46
Identity Propagation	2-48

3.	Listening for Database Events	3-1
	Understanding iWay Event Functionality	3-2
	Creating, Editing, or Deleting an Event Port	3-2
	Creating an Event Port From the iWay Events Tab	3-2
	Editing and Deleting an Event Port	3-16
	Creating, Editing, or Deleting an Event Channel	3-17
	Creating a Channel	3-17
	Editing or Deleting a Channel	3-25
	Choosing a Listening Technique	3-25
	Standard Event Processing With Row Tracking	3-27
	Standard Event Processing With Row Removal	3-33
	Trigger-based Event Processing	3-36
4.	Using Web Services Policy-Based Security	4-1
	iWay Business Services Policy-Based Security	4-2
	Configuring iWay Business Services Policy-Based Security	4-3
5.	Management and Monitoring	5-1
	Managing and Monitoring Services and Events Using iBSE	5-2
	Managing and Monitoring Services and Events Using the JCA Test Tool	5-17
	Setting Engine Log Levels	5-21
	Configuring Connection Pool Sizes	5-23
	Migrating Repositories	5-23
	File Repositories	5-24
	iBSE Repositories	5-24
	JCA Repositories	5-28
	Migrating Event Handling Configurations	5-28
	Exporting or Importing Targets	5-32
	Retrieving or Updating Web Service Method Connection Information	5-36
	Starting or Stopping a Channel Programmatically	5-40
A.	JDBC Drivers	A-1
	Copying and Collecting a JDBC File	A-2

CHAPTER 1

Introducing the iWay XML Adapter for RDBMS

Topics:

- Introduction
- Using the Adapter With Relational Databases
- Using the Adapter to Access Non-relational Databases
- Deployment Information for the Adapter

The following topics provide an overview of the iWay XML Adapter for RDBMS and how it works, including descriptions of key features and functionality.

Introduction

Since most custom and packaged applications are built with relational databases, relational database management systems (RDBMS) must be taken into consideration in an enterprise integration strategy. The iWay XML Adapter for RDBMS incorporates in-depth knowledge of relational database query access and transaction, replication, and copy management technologies to optimize the use of databases with enterprise application systems.

You also can use the adapter to gain relational database access to non-relational data sources. To access non-relational data, the adapter works in conjunction with an adapter server component that is installed and runs outside of the target database management system. For the purposes of this manual, the adapter server component is equivalent to an RDBMS.

Using the Adapter With Relational Databases

The adapter enables integration with RDBMS systems by one of the following ways:

- **Sending a Request.** The adapter sends requests to the database, using either SQL queries or stored procedures.
- **Using a Listener.** The adapter listens for database table activity.

In both cases, the query or stored procedure call is expressed to the adapter in the form of an XML document.

Key features of the adapter include:

- Asynchronous, bi-directional message interactions between applications and databases, including IBM DB2, IBM Informix, Microsoft SQL Server, Oracle, IDMS, VSAM, IMS/DB, ADABAS, Sybase RDBMSs, and others.
- iWay Application Explorer, which uses metadata on database servers to build XML schemas for use by adapter requests.
- Integration of requests and table event (outbound) operations in workflows.
- JDBC™ 2.0 standard SQL operations (DELETE, INSERT, SELECT, and UPDATE) and the execution of stored procedures against DB2, Informix, MS SQL Server, Oracle, Sybase, and any database management system accessible by the server component.
- Oracle object-relational extensions, such as processing of nested tables and arrays in accessing PL/SQL stored procedures or supporting outbound database rows on Oracle AQ queues.

Using the Adapter to Send a Request

The adapter can process SQL statements embedded in XML documents and forward them to an RDBMS (or server component) as a request. The RDBMS returns the data to the adapter, which returns the data to the client.

The adapter can:

- Receive a service request from an external client.
- Transform an XML request document into the RDBMS data format.

The request document conforms to the XML schema generated by Application Explorer and based on RDBMS metadata.

- Send the request to the RDBMS and wait for its response.
- Transform the response from the RDBMS data format to an XML document.

The XML document conforms to the XML schema for the response that was generated by Application Explorer and based on RDBMS metadata.

Using the Adapter Listener

The adapter supports the capture of events from applications that write to a database. It captures the data and performs operations based on the content of table rows. The adapter reads one or more rows from the table and creates an XML document representing the column data in each row.

Additional business logic facilities can be applied to the constructed XML document, including transformation, validation, security management, and application processing. Transformations by business logic can include deleting rows or altering column values. The resulting XML is formatted and sent to the adapter for further processing.

The listener can:

- Monitor data changes by repeatedly performing an SQL query.

The SQL listener also supports customized user exits with Java™ classes to define custom operations on the row sets.

- Be configured to operate one row at a time or to operate on sets of data.

You can configure the listener to send events only to a business process workflow when a specified minimum number of rows become available in the source RDBMS.

- Allow the configuration of an optional SQL post-query statement.

The statement performs specific RDBMS operations after the adapter sends the row set (formatted as XML) to a business process workflow.

The default operation is to delete the rows that were transferred to the workflow.

Other options can include moving the rows to an archive table or marking the rows with an SQL UPDATE.

- Support complex configurations.

For example, you may want to extract information periodically from a base table and incorporate reference data from an additional table. Records cannot be deleted from the base table and reference table.

In this case, the adapter uses a temporary table to maintain the sequenced rows in the base table. The temporary table contains a starting value for the sequence. It holds the last value of the sequence field processed by the RDBMS listener, enabling multiple event operations to collect updates while avoiding sending duplicates to a business process workflow.

- Support user-defined exits.

User-defined exits can be implemented to enable more complex programming or external database operations.

For example, an operating system program can be executed to facilitate an import or export process within a custom application.

Using the Adapter to Access Non-relational Databases

iWay Software uses a proprietary metadata management and creation tool that enables all databases on all platforms to look and act as if they were relational databases. This enables a single, uniform approach to data access.

Depending on which databases you have licensed, the following table lists several of the non-relational databases that can be accessed through the adapter.

IBM-Compatible Mainframes (MVS/VM)	OpenVMS	UNIX-Based Computers
ADABAS	ADABAS/C	ADABAS/C
CA-Datcom/DB	DBMS	C-ISAM
DB2	FOCUS	DB2/6000
FOCUS	Ingres	Essbase
CA-IDMS/DB	Oracle	FOCUS
CA-IDMS/SQL	Rdb	Interplex (DMS/RDMS 2200/1100)
IMS/DB	RMS	Informix
ISAM	Sybase	Ingres
Millennium	Progress	Oracle
MODEL 204		Progress
NOMAD		Red Brick
Oracle		Sybase ASE
SQL/DS		Sybase IQ
Supra		Teradata
System 2000		UniVerse (PICK)
Teradata		
TOTAL		
QSAM		
VSAM		

OS/400	Tandem	Windows
FOCUS SQL/400	Enscribe FOCUS NonStop SQL	ADABAS/C DB2/2 Essbase FOCUS Informix Interplex (DMS/RDMS 2200/1100) Oracle Microsoft SQL Server Microsoft Analytical Engine Sybase ASE Sybase IQ Teradata

To enable this access, iWay Software adapter structure is twofold. All Java-based adapters such as the adapters for RDBMS, IMS/TM, and CICS are hosted within an iWay Adapter Framework (iWAF) on a server platform such as Sun Java System Application Server.

A separate iWay server component hosts all of the data adapters that access the underlying non-relational data using select statements. This server component runs outside of the adapter host (for example, Sun Java System) environment. The adapter connects to the iWay data adapters hosted in the server component using a Java-based connection.

Because the server component looks as if it were a relational database, the connection string to it is the same as to any relational database, for example, to an Oracle database. Therefore, the RDBMS connections are configured similarly as to a relational database.

After you configure an adapter and create metadata using Application Explorer, you can access the database or file system (such as VSAM) using standard JDBC calls. Therefore, you can access all databases and file systems, whether mainframe, AS400, or UNIX, as if the database were a full JDBC client RDBMS after you configure them on the server.

Read access is supported by all iWay adapters. Write access is supported by all relational adapters such as DB400 and OS390 DB2. Some adapters do not support write access, for example, CA-IDMS/DB, Datacom, and Model 204. Read/write access is supported by ADABAS, VSAM, and IMS via SQL insert and update statements. Depending on the type of database accessed, the server component could have specific platform requirements. For the applicable database in question, see the iWay documentation.

In the usual non-relational database access scenario, the iWay XML Adapter for RDBMS (hosted, for example, by the Sun Java System platform) connects to the iWay Adapter for VSAM (hosted by the server component) using JDBC standards. Application Explorer is used to configure this connection. You can then create Web services for SQL and parameterized SQL using Application Explorer. You also can use Application Explorer to create events that occur within the database, such as an insert to a VSAM file or a modification of a VSAM record.

Accessing Stored Procedures for Non-relational Data Sources

The adapter is used when there is a specific requirement to create and execute catalogued iWay stored procedures (remote procedure calls, also referred to as RPCs) on the server component. iWay uses a very powerful fourth generation language that is much more robust than SQL.

iWay stored procedures on the server component can be created to enable complex multi-platform joins, specialized routines, and so forth. iWay stored procedures also enable COBOL or RPG programs to be executed. To use the adapter, an iWay stored procedure must be catalogued before Application Explorer can create the schemas or Web services for that stored procedure. For more information on using the extended functionality within iWay stored procedures, contact iWay Customer Support Services.

When used in conjunction with Application Explorer, the adapter creates Web services that can be used to run the stored procedures from any Web client.

Deployment Information for the Adapter

- The adapter works in conjunction with Application Explorer and the iWay server component (for non-relational database access only) and one of the following components:
- iWay Business Services Engine (iBSE)
- iWay Enterprise Connector for J2EE™ Connector Architecture (JCA)

Application Explorer, used to configure database connections and create Web services and events, can be configured to work in a Web services environment in conjunction with the iWay Business Services Engine or with the iWay Enterprise Connector for J2EE Connector Architecture (JCA). When working in a JCA environment, the connector uses the Common Client Interface (CCI) to provide fast integration services using iWay Adapters instead of using Web services.

Both iBSE and the iWay Connector for JCA are deployed to the Sun Java System environment with Application Explorer and the adapters. The iWay server component is deployed outside of the adapter host (for example, Sun Java System Application Server) and outside of the target database management system.

Deployment Information Roadmap

The following table lists the location of deployment information for the adapter. A description of the iWay Business Services Engine (iBSE) and the iWay Enterprise Connector for J2EE Connector Architecture (JCA) follow the table.

Deployed Component	For more information, see
iWay Application Explorer	Chapters 2 and 3 of this guide <i>iWay Installation and Configuration</i>
iWay Business Services Engine (iBSE)	<i>iWay Installation and Configuration</i>
iWay Enterprise Connector for J2EE Connector Architecture (JCA)	<i>iWay Connector for JCA User's Guide</i> <i>iWay Installation and Configuration</i>
iWay server component	<i>iWay Server Installation</i> <i>iWay Server Administration for UNIX, Windows, OpenVMS, OS/400, OS/390, and z/OS</i> <i>iWay Data Adapter Administration for UNIX, Windows, OpenVMS, OS/400, OS/390, and z/OS</i>

The iWay Business Services Engine (iBSE)

The iWay Business Services Engine (iBSE) exposes—as Web services—enterprise assets that are accessible from adapters regardless of the programming language or the particular operating system.

iBSE simplifies the creation and execution of Web services when running:

- Custom and legacy applications
- Database queries and stored procedures
- Packaged applications
- Terminal emulation and screen-based systems
- Transactional systems

Web services is a distributed programming architecture that solves Enterprise Application Integration (EAI) hurdles that other programming models cannot. It enables programs to communicate with one another using a text-based but platform and language independent message format called XML.

Coupled with a platform and language independent messaging protocol called SOAP (Simple Object Access Protocol), XML enables application development and integration by assembling previously built components from multiple Web services.

The iWay Enterprise Connector for J2EE Connector Architecture (JCA)

The iWay Enterprise Connector for J2EE Connector Architecture (JCA) enables developers of JCA-compliant applications to deploy iWay adapters as JCA resources. The connector is supported on the Sun Java System Application Server.

The iWay Connector for JCA is distributed as a standard Resource Adapter Archive (RAR) for deployment to the application server. Thus, the connector can be used in systems that are non-compliant, although services such as pooled connections are not available.

Two connectors are distributed in the iWay installation package. One conforms to the JCA 1.0 specification, with extensions that allow for the consumption of events. The other conforms to the JCA 1.5 specification. The JCA 1.0 connector provides for event functionality through the configuration of ports and channels. When using the adapter in conjunction with a JCA 1.5 connector, there is no need to create event ports to dispose of event data. However, you must create a channel to enable event listening capabilities. For more information on event capabilities of the iWay JCA connectors, see Chapter 3, *Listening for Database Events*. For more information on installing and deploying both connectors see *iWay Installation and Configuration*.

Deployment Information for the Adapter

CHAPTER 2

Creating XML Schemas or Business Services

Topics:

- Generating Schemas and a Business Service
- Starting Servlet Application Explorer
- Creating and Managing a Connection
- Viewing Metadata
- Creating a Statement and Generating Schemas
- Request and Response Documents
- Multiple Executions of the Same SQL Statement or Stored Procedure
- Stored Procedures with Constraints
- Understanding iWay Business Services

This section describes how to use Application Explorer to:

- View metadata that describes your SQL statements and stored procedures.
- Create SQL statements and generate XML schemas that define request and response documents.
- Create business services (also known as Web services) for your SQL statements and stored procedures.

Generating Schemas and a Business Service

You can use Application Explorer to connect to relational databases, such as Oracle or Informix and to non-relational databases, such as an IMS database.

When connected to non-relational databases, the adapter uses an iWay server component which enables you to take advantage of all the integration capabilities offered by iWay Software for mainframe database access. For example, when using the iWay server component, you gain relational database access to non-relational data. Stored procedures for non-relational databases can be created and stored using the server component.

Procedure: How to Generate Request and Response Schemas or a Business Service

To generate request and response document schemas or a business service:

1. Start iWay Servlet Application Explorer.

You can open a new or existing connection to a relational or non-relational database management system, as described in *Creating and Managing a Connection* on page 2-4.

After you connect to a system, you can expand the iWay Adapters node to view the list of adapters installed on your system. After you finish using a connection, you can close it. If you will not need the connection in the future, you can delete it.

Note: When you close Application Explorer, it automatically closes all open connections.

2. Generate XML schemas.

The schemas define request and response documents for your SQL statements and stored procedures, as described in *Creating a Statement and Generating Schemas* on page 2-14.

You can use the schemas when you create request documents and when you develop logic to process responses.

3. Create request documents.

You create documents for each operation against each table and for each stored procedure. You can use a third-party XML tool to generate a request document from the XML schema.

You also may want to examine the metadata describing your SQL statements and procedures, as described in *Viewing Metadata* on page 2-11. For information about request and response document formats, see *Request and Response Documents* on page 2-24.

4. Generate a business service (also known as a Web service) for an SQL statement or stored procedure. For more information on Web services, see *Understanding iWay Business Services* on page 2-41.

Starting Servlet Application Explorer

The server must be started where Servlet Application Explorer is running.

Procedure: How to Start Application Explorer

1. Ensure the server is started where Application Explorer is running.
2. Enter the following URL in your browser window:

`http://hostname:port/iwae/index.html`

where:

`hostname`

Is the machine where Application Explorer is installed.

`port`

Is the port number for iBSE. The default port is 80.

Application Explorer opens. The Available Hosts drop-down list appears in the upper-right corner. Three tabs appear near the top of the Application Explorer screen. From left to right they are:

- iWay Adapters, where you create and manage connections to RDBMS.
- iWay Events, where you configure RDBMS event listening.
- iWay Business Services, where you create and view business services.

The left pane of the window contains an expandable list of adapter nodes (based on the adapters installed), events, or business services, depending on the tab that is selected. The right pane provides the details of the selected adapter, event, or service and is the work area where you define and modify adapter functions and services.

The Available Hosts drop-down list specifies to which Servlet iBSE instance or JCA instance you connect.

For more information on accessing different instances of a JCA installation or a Servlet iBSE, see the *iWay 5.5 Installation and Configuration* documentation.

Creating and Managing a Connection

To access an adapter, you must define a target that connects to the adapter. After the defined target is created, it automatically is saved. You must establish a connection to the defined target every time you start Application Explorer or after disconnecting.

Procedure: How to Define a New Target

To define a new target:

1. In the left pane of Application Explorer, expand the *iWay Adapters* node.
2. Click the *RDBMS* node.
3. In the right pane, move the pointer over *Operations* and select *Define a new target*.

The Add a new RDBMS target pane opens on the right with a definition of targets and instructions for adding a target.

Add a new RDBMS target

Targets represent configured connections to instances of backend systems. Choose a name and description for the new target that you wish to create.

Target Name:

Description:

Target Type:

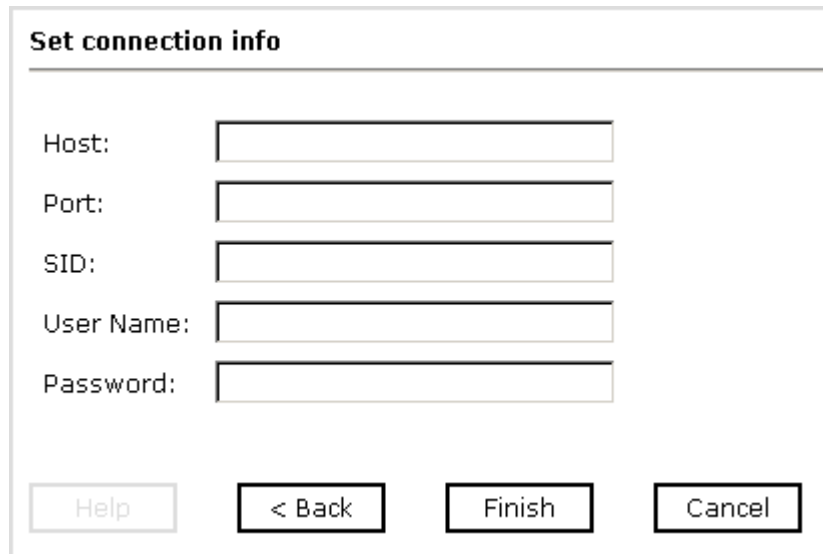
- a. In the Target Name field, type a descriptive name for the target, for example, NewTarget.
- b. In the Description field, type a brief description for the connection.
- c. From the Target Type drop-down list, select the type of target to connect to.

Important: Oracle is the default value. If you are connecting to a non-relational database, select *iWay Server*. This establishes a connection to the iWay server component. The server component is used to provide relational database access to non-relational databases. For more information on the server component, see Chapter 1, *Introducing the iWay XML Adapter for RDBMS*.

Note: The EDA Server and WebFOCUS Server target options are for connecting to Information Builders resources. The types of databases that display in the Target Type drop-down list depend on the JAR files you have installed in your lib directory.

4. Click Next.

The Set connection info dialog box opens in the right pane and includes fields to complete to set your connection parameters. The fields that appear in the Set connection info pane are specific to the type of database to which you are connecting.



The image shows a dialog box titled "Set connection info". It contains five text input fields, each with a label to its left: "Host:", "Port:", "SID:", "User Name:", and "Password:". Below the input fields are four buttons: "Help", "< Back", "Finish", and "Cancel".

Note: The RDBMS connection parameters are consistent with those found in your RDBMS system. For more information on parameter values that are specific to your RDBMS configuration, consult your RDBMS system administrator.

5. Enter connection information that is specific to the database to which you want to connect.

The following table lists and defines connection parameters.

Parameter	Definition
Host	DNS or IP name of the server where the database instance resides.
Port	Port number on which the database is listening.
Data Source Name	Specific name of the database or data source to which you connect.
Server Name	For an iWay server component, the name of the service node to which you are connecting.
Driver	Name of the driver used to access the database you want to connect to. For more information, see your database documentation.
Initial Context Factory	JNDI context.INITIAL_CONTEXT_FACTORY that is provided by the JNDI service provider. For Sun Java System Application Server, this is: <code>com.sun.jndi.cosnaming.CNCtxFactory</code>
URL	<p>For a JDBC connection, the JDBC driver-specific URL used to connect to the RDBMS.</p> <p>For information on using driver options in the URL definition, for example, <code>selectMethod=Cursor</code> for Microsoft SQL, see <i>Using URL Options</i> on page 2-7.</p> <p>For a data source connection, the URL to use to contact the JNDI provider. The syntax of this URL depends on the JNDI provider being used. This value corresponds to the standard JNDI property, <code>java.naming.provider.url</code></p> <p>The URL of the Sun Java System Application Server is <code>iiop://localhost:3700</code> where: <code>3700</code> Is a default port.</p>

Parameter	Definition
JNDI Name	JNDI name of a queue to which events are emitted.
Database	Oracle, SQL Server, Sybase, DB2, Informix, EDA Server, or other.
SID	For an Oracle database, the unique name of the database service selected by the database administrator or the person who installed Oracle.
User Name	Database user ID to access the database. The user ID must have database access to the interface tables that are accessed.
Password	Password associated with the specified user name.
Trace	Type 0 to disable tracing and 1 to enable tracing.
Trace File	Path and file name for the trace file.

Note: When connecting to the iWay server component for access to non-relational databases, the connection information must be the same for all databases in the server component system.

6. Click *Finish*.

In the left pane, the target name appears under the node where you created the new target. You have finished creating the new target.

For information on connecting to the target, see *How to Connect to a Defined Target* on page 2-8.

Reference: Using URL Options

You can use the `selectMethod` as a connection parameter in the URL. The parameters you add to the driver URL depends on the driver you use.

Note: Each driver has specific JDBC system properties, most of which can be used in the connection string. For more information on these properties, refer to the documentation on the specific driver.

- **For Microsoft SQL**

To avoid some exceptions when using the iWay XML Adapter for RDBMS with Microsoft SQL Server 2000 Driver for JDBC, you must add `selectMethod=cursor` to the JDBC URL specification. For example,

```
jdbc:microsoft:sqlserver://PMSNJC:1433;DatabaseName=dbname;selectMethod=cursor;
```

This statement determines whether Microsoft SQL Server "server cursors" are used for SQL queries. Because the adapter is not limited to a single active statement while executing a set of queries within a transaction, adding this statement to the JDBC URL allows you to specify multiple queries within a transaction. This helps to prevent errors because it addresses default settings in the adapter and in the driver.

The benefit of specifying this statement is that it enables you to have multiple concurrent statements open from a given connection, which is required for pooled connections.

- **For DB2**

You can include a `translate binary` option to control how binary and varbinary data values are treated. On the OS/400 system, for example, if a field is tagged with CCSID 65535, you can set an optional `translate binary` parameter to true, which instructs the JDBC driver to translate the field to EBCDIC characters; for example,

```
jdbc:db2://host:port/DatabaseName;translate binary=true
```

The `translate binary` option forces the JDBC driver to treat binary and varbinary data values as if they were char and varchar data. The default is set to false. This setting is usually needed if the columns were created using different character coding values.

- **For Sybase**

You can ensure the use of dynamic statements by using the `DYNAMIC_PREPARE` option; for example,

```
jdbc:sybase:Tds:host:port/DatabaseName?&DYNAMIC_PREPARE=true
```

By default this option is set to false

Procedure: How to Connect to a Defined Target

1. In the left pane of Application Explorer, expand the *iWay Adapters* node.

The following image shows the RDBMS node expanded with `NewTarget` selected. The `x` under the connection icon indicates that the connection is closed.



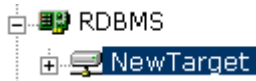
2. Expand the *RDBMS* node and select the connection you want to open (for example, *NewTarget*).
3. In the right pane, move the pointer over *Operations* and select *Connect*.
The connection pane opens and displays the connection information.
4. Verify your connection parameters. If required, provide the password and then click *OK*.
If the parameters are correct and the RDBMS component is available, the node under the RDBMS node displays a plus sign indicating that you are connected to the defined target. Otherwise, an error message appears in the right pane.

Disconnecting From a Defined Target

Although you can maintain multiple open connections, iWay Software recommends disconnecting from targets that are not in use.

Procedure: How to Disconnect From a Defined Target

1. In the left pane of Application Explorer, expand the *iWay Adapters* node.
2. Expand the *RDBMS* node.
3. The following image shows the RDBMS node expanded with *NewTarget* selected. The plus icon associated with the connection icon indicates that the connection is open.



4. Click the connection you want to close, for example, *NewTarget*.
5. In the right pane, move the pointer over *Operations* and select *Disconnect*.

Disconnecting from the application closes the connection, but the connection still appears in the left pane so that you can re-open it.

The connection node now has an x icon, indicating that it is closed, as shown in the following image.



When you want to re-establish a connection, *Connect* is available from the pop-up menu.

Editing a Defined Target

After you create a defined target using Application Explorer, you can edit any information that you provided during the creation process.

Procedure: How to Edit a Defined Target

1. In the left pane of Application Explorer, expand the *iWay Adapters* node.
2. Expand the *RDBMS* node.
3. Click the connection you want to edit, for example, *NewTarget*.
4. In the right pane, move the pointer over *Operations* and select *Edit*.

The Edit pane opens on the right and includes three fields (Target Name, Description, and Target Type) and two active buttons (Next and Cancel).

Edit RDBMS target NewTarget

Targets represent configured connections to instances of backend systems. Choose a name and description for the new target that you wish to create.

Target Name:

Description:

Target Type:

5. Modify the target information as required and then, click *Next*.
The Set connection info pane opens on the right.
6. Modify the information as required and then, click *Finish*.

Deleting a Defined Target

You can delete a target, rather than just disconnecting and closing it. When you delete the target, the node disappears from the list of RDBMS targets in the left pane of the explorer.

Procedure: How to Delete a Defined Target

1. In the left pane of Application Explorer, expand the *Way Adapters* node.
2. To view the list of connections, expand the *RDBMS* node.
3. Click the connection you want to delete.
4. In the right pane, move the pointer over *Operations* and select *Delete*.

A message appears, prompting you to confirm the deletion of the node.

5. Click *OK*.

The node disappears from the list of available connections.

Viewing Metadata

Viewing metadata is useful when creating request documents. You can view:

- Table metadata, as described in *How to View Table Metadata on page 2-11*.
- Stored procedure metadata for a relational or a non-relational database, as described in *How to View Stored Procedure Metadata on page 2-13*.

Procedure: How to View Table Metadata

To view table metadata:

1. If you are not connected to a defined target, connect to one, as described in *How to Connect to a Defined Target on page 2-8*.
2. Expand the *Schemas* node under the desired connection.
3. Select a database.
4. Expand the *Tables* node.
5. Scroll down and select a table to view.

Note: Although the list of tables includes all tables in the RDBMS, the user ID you specified for the connection may not have access to the table you selected. If this is the case, the creation of schemas fails.

When you select a table, a metadata summary table appears in the right pane containing Property and Value columns. The Value column contains a description of the table and an ellipsis symbol that you can click to access database properties and columns as shown in the following image.

Properties for empdata

Property	Value
iwaf.description	
Database Properties	...
Columns	...

- a. Depending on which properties you want to view, click the ellipsis symbol in the Database Properties or the Columns row.

The properties (for example, data type and column size, represented by columns in a table) appear in the right pane as shown in the following image.

Details for collection property Columns

column name	data type	type name	column size	buffer length	decimal digits	num prec radix	nullable
PIN	1	CHAR	9		null	null	0
LASTNAME	1	CHAR	15		null	null	0
FIRSTNAME	1	CHAR	10		null	null	0
MIDINITIAL	1	CHAR	1		null	null	0
DIV	1	CHAR	4		null	null	0
DEPT	1	CHAR	20		null	null	0
JOBCLASS	1	CHAR	8		null	null	0
TITLE	1	CHAR	20		null	null	0
SALARY	8	DOUBLE	15		2	10	0
HIREDATE	9	DATE	10		null	null	0
AREA	1	CHAR	13		null	null	0

Close

- b. When you are ready to create a schema, use this information to determine the table (or tables) and fields to use.

Procedure: How to View Stored Procedure Metadata

To view stored procedure metadata:

1. If you are not connected to a defined target, connect to one, as described in *How to Connect to a Defined Target* on page 2-8.
2. Expand the *Schemas* node under the desired connection.
3. Select a database.
4. Expand the *Procedures* node.
5. Scroll down and select the procedure to view.

Note: Although the list of procedures includes all procedures in the database, the user ID specified for the connection may not have access to the specified procedure. If this is the case, the creation of schemas fails.

When you select a procedure, a metadata summary table appears in the right pane containing Property and Value columns. The Value column contains a description of the procedure and an ellipsis symbol that you can click to access database properties and columns as shown in the following image.

Properties for empdata

Property	Value
iwaf.description	
Database Properties	...
Columns	...

- a. Depending on the properties you want to view, click the ellipsis symbol in the Database Properties or the Columns row.

The properties (for example, data type and column size, represented by columns in a table) appear in the right pane as shown in the following image.

Details for collection property Columns

column name	data type	type name	column size	buffer length	decimal digits	num prec radix	nullable
PIN	1	CHAR	9		null	null	0
LASTNAME	1	CHAR	15		null	null	0
FIRSTNAME	1	CHAR	10		null	null	0
MIDINITIAL	1	CHAR	1		null	null	0
DIV	1	CHAR	4		null	null	0
DEPT	1	CHAR	20		null	null	0
JOBCLASS	1	CHAR	8		null	null	0
TITLE	1	CHAR	20		null	null	0
SALARY	8	DOUBLE	15		2	10	0
HIREDATE	9	DATE	10		null	null	0
AREA	1	CHAR	13		null	null	0

Close

- b. Use this information to determine the procedure (or procedures) and fields to use when you are ready to create a schema.

Creating a Statement and Generating Schemas

You can create an SQL statement even when using the adapter for non-relational databases. After you create the statement, you can generate schemas that define request and response documents. The metadata is stored in the iWay Repository, which can be implemented in an RDBMS (such as Oracle or Microsoft SQL Server), a file system, or a specialized XML database. You can generate the following types of statements:

- Regular SQL Statements, as described in *How to Create a Regular SQL Statement* on page 2-15.
- Parameterized SQL statements, as described in *How to Create a Parameterized SQL Statement* on page 2-17.

- Batch statements, as described in *How to Create a Batch Statement* on page 2-21.

You can generate request and response schemas for:

- Regular (non-parameterized) SQL statements and parameterized SQL statements, as described in *How to Generate a Schema for a Prepared Statement* on page 2-22.
- Stored procedures for relational databases, and iWay stored procedures for non-relational databases, as described in *Generating a Schema for a Stored Procedure* on page 2-23.

When deployed to the Sun Java System Application Server and used in conjunction with an iBSE configuration, Application Explorer stores the schemas it creates in subdirectories under the Sun directory structure on the machine where the Sun Java System Application Server is installed. When used in conjunction with an iWay JCA configuration, Application Explorer stores schemas in a subdirectory of the iWay home directory.

When using the adapter with an iBSE configuration, the schemas are stored under a \schemas subdirectory in the Sun Java System Application Server installation directory, for example,

```
C:\SUN\AppServer\domains\domain1\applications\j2ee-apps\ibse\ibse_war\wsd1\schemas\service\RDBMS\NewTarget
```

where:

NewTarget

Is the name of the connection to the RDBMS system as defined in Application Explorer. Under this directory, Application Explorer creates subdirectories containing schemas.

When using the adapter with a JCA configuration, the schemas are stored under a \schemas subdirectory of the iWay home directory, for example,

```
C:\Program Files\iWay55\config\base\schemas\RDBMS\NewTarget
```

where:

NewTarget

Is the name of the connection to the RDBMS system as defined in Application Explorer. Application Explorer stores the schemas in this directory.

If you plan to create business services, you are not required to generate a schema. For more information, see *Understanding iWay Business Services* on page 2-41.

Procedure: How to Create a Regular SQL Statement

To create an SQL statement:

1. If you are not connected to a defined target, connect to one, as described in *How to Connect to a Defined Target* on page 2-8.

2. Click the *Statements* node.
3. In the right pane, move the pointer over *Operations* and select *Create Prepared Statement*.

The Create Prepared Statement input area opens in the right pane as shown in the following image.

Create Prepared Statement

Name :

Enter SQL Statement :

- a. In the Name field, type a name for the statement.
iWay Software recommends that you specify a name that describes the service. For example, a name of CustomerIntField could represent a request against the Customer Interface table returning a Field format response document.
 - b. In the Enter SQL Statement field, type the SQL statement for the adapter to use.
Note: If you are not the owner of the table(s), the table name must be fully qualified.
4. Click *Create*.

After the SQL statement node is built, you are ready to test the statement.

- For information on testing a regular SQL statement, see *How to Test an SQL Statement* on page 2-17.
- For information on creating schemas for both parameterized and regular SQL statements, see *How to Generate a Schema for a Prepared Statement* on page 2-22.

Procedure: How to Test an SQL Statement

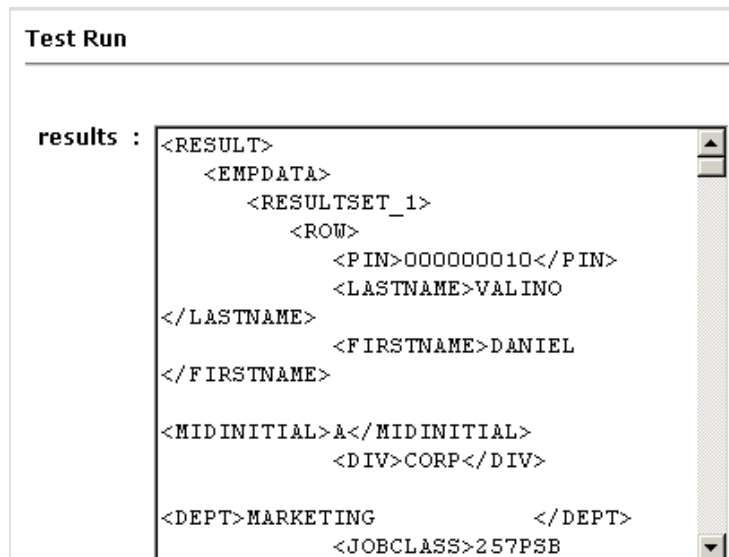
To test an SQL statement:

1. Select the SQL statement node you want to test.
2. In the right pane, move the pointer over *Operations* and select *Test Run*.

The Test Run pane opens on the right.

3. Click *Test*.

The results appear in the Test Run results window as shown in the following image.



The screenshot shows a window titled "Test Run" with a "results :" label. The results are displayed as XML text within a scrollable area. The XML structure is as follows:

```

<RESULT>
  <EMPDATA>
    <RESULTSET_1>
      <ROW>
        <PIN>000000010</PIN>
        <LASTNAME>VALINO
      </LASTNAME>
        <FIRSTNAME>DANIEL
      </FIRSTNAME>
        <MIDINITIAL>A</MIDINITIAL>
        <DIV>CORP</DIV>
        <DEPT>MARKETING           </DEPT>
        <JOBCLASS>257PSB
  
```

4. To exit the results window, click *OK*.

Creating and Testing a Parameterized SQL Statement

Parameterized SQL allows an SQL statement to be stored within the repository system with parameters imbedded within it. These parameters can be retrieved from XML documents at run time and executed against the SQL statements specified at design time. Application Explorer creates and maps parameters for the parameterized SQL at design time.

Procedure: How to Create a Parameterized SQL Statement

To create a parameterized SQL statement:

If you are not connected to a defined target, connect to one, as described in *How to Connect to a Defined Target* on page 2-8.

1. Click the *Statements* node.

2. In the right pane, move the pointer over *Operations* and select *Create Prepared Statement*.

The Create Prepared Statement pane opens on the right as shown in the following image.

Create Prepared Statement

Name :

Enter SQL Statement :

- a. In the Name field, type a name for the statement.
- b. In the Enter SQL Statement field, type the parameterized SQL statement.

Note: If you are not the owner of the table(s), the table name must be fully qualified.

3. Click *Create*.

The Parameter Data Type selection information appears in the right pane as shown in the following image.

Create Prepared Statement

params

Parameter Name	Data Type
<input type="text" value="param0"/>	UNKNOWN <input type="button" value="v"/>

- a. In the Parameter Name column, type a name for each parameter.
 - b. In the Data Type column, select a data type for each parameter from the drop-down list.
4. Click *Update*.

The properties table for the newly created statement appears in the right pane containing Property and Value columns. The Value column contains a description of the SQL statement, the actual SQL statement, and ellipsis symbols you can click to access parameters and database properties.

Operations ▶
Properties for ParamSQL2

Property	Value
iwaf.description	
Statement	Select * from empdata where LASTNAME=?
Parameters	...
Database Properties	...

5. Depending on which properties you want to view, click the ellipsis symbol in the Parameters or Database Properties row.

For information on testing a parameterized SQL statement, see *How to Test a Parameterized SQL Statement on page 2-19*.

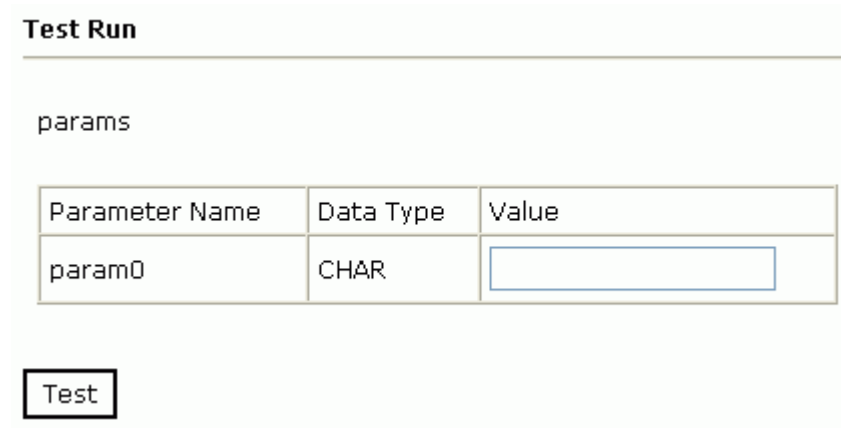
For information on creating schemas for both parameterized and regular SQL statements, see *How to Generate a Schema for a Prepared Statement on page 2-22*.

Procedure: How to Test a Parameterized SQL Statement

To test a parameterized SQL statement:

- 1. Select the parameterized SQL statement node you want to test.
- 2. In the right pane, move the pointer over *Operations* and select *Test Run*.

The Test Run pane opens on the right for the SQL statement containing the parameter name, data type, and an input box where you can type the parameter value as shown in the following image.



The image shows a 'Test Run' pane with a table and a button. The table has three columns: 'Parameter Name', 'Data Type', and 'Value'. The first row contains 'param0', 'CHAR', and an empty input box. Below the table is a 'Test' button.

Parameter Name	Data Type	Value
param0	CHAR	<input type="text"/>

3. For each parameter, type a value in the Value field.

For example, provide a sample character value, for example, BELLA, for the following SQL statement:

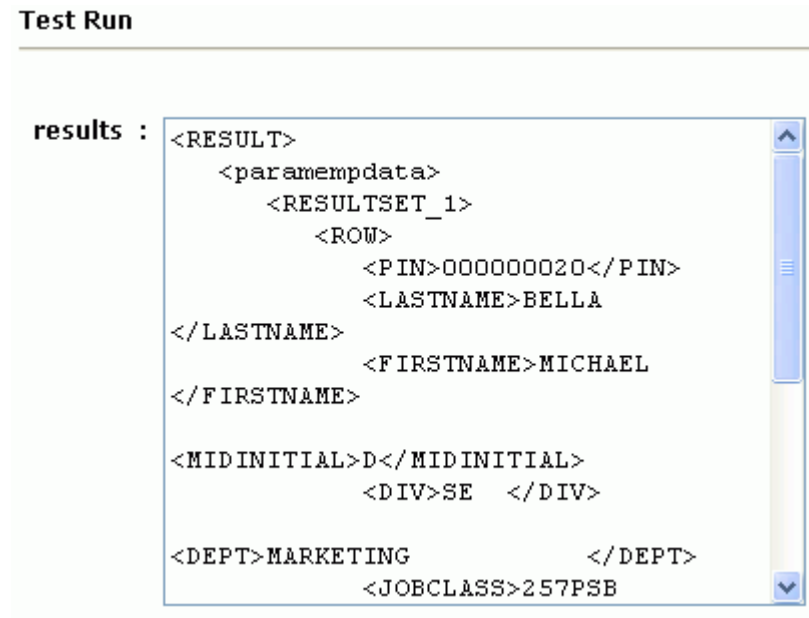
```
Select * from empdata where LASTNAME=?
```

In this example, the values correspond to values of fields found in a table.

Parameterized statements may include parameters that are input for SQL functions, for example, the Oracle SQL function TO_DATE(StringParm). In this case, the data type selected is the expected data type of the SQL function. This is why you provide the SQL type when you create the prepared parameterized SQL statement.

4. Click *Test*.

The results appear in the Test Run results window as shown in the following image.



5. To exit the results, click *OK*.

Creating a Batch Statement

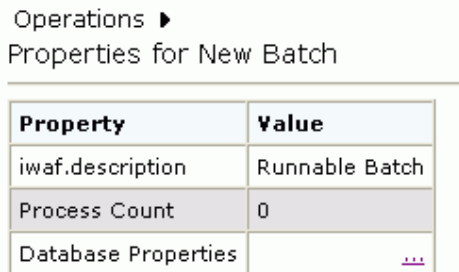
Batch statements enable you to execute multiple SQL and/or parameterized SQL statements simultaneously.

Procedure: How to Create a Batch Statement

To create a batch statement:

1. If you are not connected to a defined target, connect to one, as described in *How to Connect to a Defined Target* on page 2-8.
2. In the left pane, select the *Batches* node.
3. In the right pane, move the pointer over *Operations* and select *Create A Batch*.
The Create A Batch pane opens on the right.
4. Type a name for the new Batch and click *Create*.

The batch properties information appears in the right pane containing Property and Value columns. The Value column contains a description of the batch, the process count, and an ellipsis symbol that enables you to access database properties as shown in the following image.



The screenshot shows a software interface with a menu titled 'Operations' and a sub-menu 'Properties for New Batch'. Below the menu is a table with two columns: 'Property' and 'Value'. The table contains three rows: 'iwaf.description' with value 'Runnable Batch', 'Process Count' with value '0', and 'Database Properties' with an ellipsis symbol '...' in the Value column.

Property	Value
iwaf.description	Runnable Batch
Process Count	0
Database Properties	...

5. Move the pointer over *Operations* and choose whether to add stored procedures or statements.

The Add Statement or Add Procedure drop-down selection appears in the right pane.

6. From the drop-down list, select the statement or procedure and click *Next*.
7. To add more statements or procedures, select the batch node in the left pane, and then select the appropriate option from the Operations menu in the right pane.

Generating a Schema for a Prepared Statement

You must first create the prepared statement before generating a schema for it. For more information on creating prepared statements, see the following procedures:

- *How to Create a Regular SQL Statement* on page 2-15.
- *How to Create a Parameterized SQL Statement* on page 2-17.
- *How to Create a Batch Statement* on page 2-21.

Procedure: How to Generate a Schema for a Prepared Statement

To generate a schema for a prepared statement:

1. If you are not connected to a defined target, connect to one, as described in *How to Connect to a Defined Target* on page 2-8.
2. Click the *Statements* node.
3. Click the node containing the prepared statement for which you want to generate a schema.
4. In the right pane, move the pointer over *Operations* and select *Generate Schema*.

A table that lists the available schemas appears.

5. To view a schema, click the ellipsis symbol in the Schema column.

The schema is generated and ready to use. You can use the generated request schema to create a sample XML document to be used by the adapter. To add a schema to a business service, see *Understanding iWay Business Services* on page 2-41.

Generating a Schema for a Stored Procedure

The following procedure describes how generate a schema for stored procedures for relational databases and how to generate a schema for iWay stored procedures for non-relational databases.

Procedure: How to Generate a Schema for a Stored Procedure

To generate a schema for a stored procedure:

1. If you are not connected to a defined target, connect to one, as described in *How to Connect to a Defined Target* on page 2-8.
2. Expand the *Schemas* node under the desired connection.
3. Select the database containing the stored procedure for which you want to generate a schema.
4. Expand the *Procedures* node.
5. Select the stored procedure.
6. In the right pane, move the pointer over *Operations* and select *Generate Schema*.

The Schemas table appears in the right pane containing three columns named Part, Root Tag, and Schema and three rows named Request, Response, and Event as shown in the following image.

Schemas		
Part	Root Tag	Schema
Request	RDBMS	...
Response	RESULT	...
Event	N/A	N/A

- a. To view the request schema, click the ellipsis symbol that is located in the third column of the Request row.
- b. To view the response schema, click the ellipsis symbol that is located in the third column of the Response row.

The schemas are now ready to use. You can use the generated request schema to create a sample XML document to be used by the adapter.

7. Click *OK*.

Request and Response Documents

You can generate request document schemas using Application Explorer, as described in *Creating a Statement and Generating Schemas* on page 2-14. You can generate request document instances using a third party XML tool and submit those documents to the RDBMS or iWay agent.

The following topics include examples of schemas and instance documents for:

- Regular SQL Statements
- Parameterized SQL Statements
- Stored Procedure Schemas for an Oracle Database
- Stored Procedure Schemas for a VSAM Database

Regular SQL Statements

The following examples are based on schemas created for a regular SQL statement.

Example: Regular SQL Request Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Generated by the iBSE 2004-01-13T22:22:33Z -->
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="RDBMS">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="stock_price_select">
          <xsd:complexType>
            <xsd:attribute name="location" type="xsd:string"
use="optional" fixed="RDBMS/Statements/stock price select"/>
          </xsd:complexType>
        </xsd:element>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

Example: Regular SQL Request Instance Document

```
<?xml version="1.0" encoding="UTF-8"?>
<RDBMS xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="D:\iway\RDBMS\stock price
select_request.xsd">
  <stock_price_select location="RDBMS/Statements/stock price select"/>
</RDBMS>
```

Example: Regular SQL Response Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Generated by the iBSE 2004-01-13T22:22:33Z -->
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="RESULT">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="stock_price_select">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="RESULTSET_1" minOccurs="0">
                <xsd:complexType>
                  <xsd:sequence>
                    <xsd:element name="ROW" minOccurs="0" maxOccurs="unbounded">
                      <xsd:complexType>
                        <xsd:sequence>
                          <xsd:element name="RIC" type="xsd:string"/>
                          <xsd:element name="PRICE" type="xsd:double"/>
                          <xsd:element name="UPDATED" type="xsd:date"/>
                          <xsd:element name="RR" type="xsd:double"/>
                        </xsd:sequence>
                      </xsd:complexType>
                    </xsd:element>
                  </xsd:sequence>
                </xsd:complexType>
              </xsd:element>
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

Example: Regular SQL Response Instance Document

```
<?xml version="1.0" encoding="UTF-8"?>
<RESULT xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="D:\iway\RDBMS\stock price
select_response.xsd">
  <stock_price_select>
    <RESULTSET_1>
      <ROW>
        <RIC>String</RIC>
        <PRICE>3.14159265358979</PRICE>
        <UPDATED>1967-08-13</UPDATED>
        <RR>3.14159265358979</RR>
      </ROW>
      <ROW>
        <RIC>String</RIC>
        <PRICE>3.14159265358979</PRICE>
        <UPDATED>1967-08-13</UPDATED>
        <RR>3.14159265358979</RR>
      </ROW>
      <ROW>
        <RIC>String</RIC>
        <PRICE>3.14159265358979</PRICE>
        <UPDATED>1967-08-13</UPDATED>
        <RR>3.14159265358979</RR>
      </ROW>
    </RESULTSET_1>
  </stock_price_select>
</RESULT>
```

Parameterized SQL Statements

The following examples are based on schemas created for a parameterized SQL statement.

Example: Parameterized SQL Request Statement

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Generated by the iBSE 2004-01-13T21:57:19Z -->
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="RDBMS">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="paramempdata">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="param0">
                <xsd:complexType>
                  <xsd:simpleContent>
                    <xsd:extension base="xsd:string">
                      <xsd:attribute name="dataType"
type="xsd:string" use="required" fixed="CHAR"/>
                    </xsd:extension>
                  </xsd:simpleContent>
                </xsd:complexType>
              </xsd:element>
            </xsd:sequence>
            <xsd:attribute name="location" type="xsd:string"
use="optional" fixed="RDBMS/Statements/paramempdata"/>
          </xsd:complexType>
        </xsd:element>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

Example: Parameterized SQL Request Instance Document

```
<?xml version="1.0" encoding="UTF-8"?>
<RDBMS xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="D:\iway\RDBMS\paramempdata_request.xsd">
  <paramempdata location="RDBMS/Statements/paramempdata">
    <param0 dataType="CHAR">String</param0>
  </paramempdata>
</RDBMS>
```

Example: Parameterized SQL Response Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Generated by the iBSE 2004-01-13T21:57:20Z -->
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="RESULT">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="paramempdata">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="RESULTSET_1" minOccurs="0">
                <xsd:complexType>
                  <xsd:sequence>
                    <xsd:element name="ROW" minOccurs="0" maxOccurs="unbounded">
                      <xsd:complexType>
                        <xsd:sequence>
                          <xsd:element name="PIN" type="xsd:string"/>
                          <xsd:element name="LASTNAME" type="xsd:string"/>
                          <xsd:element name="FIRSTNAME" type="xsd:string"/>
                          <xsd:element name="MIDINITIAL" type="xsd:string"/>
                          <xsd:element name="DIV" type="xsd:string"/>
                          <xsd:element name="DEPT" type="xsd:string"/>
                          <xsd:element name="JOBCLASS" type="xsd:string"/>
                          <xsd:element name="TITLE" type="xsd:string"/>
                          <xsd:element name="SALARY" type="xsd:float"/>
                          <xsd:element name="HIREDATE" type="xsd:date"/>
                          <xsd:element name="AREA" type="xsd:string"/>
                        </xsd:sequence>
                      </xsd:complexType>
                    </xsd:element>
                  </xsd:sequence>
                </xsd:complexType>
              </xsd:element>
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```


Example: Parameterized SQL Response Instance Document

```

<?xml version="1.0" encoding="UTF-8"?>
<RESULT xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="D:\iway\RDBMS\paramempdata_response.xsd">
  <paramempdata>
    <RESULTSET_1>
      <ROW>
        <PIN>String</PIN>
        <LASTNAME>String</LASTNAME>
        <FIRSTNAME>String</FIRSTNAME>
        <MIDINITIAL>String</MIDINITIAL>
        <DIV>String</DIV>
        <DEPT>String</DEPT>
        <JOBCLASS>String</JOBCLASS>
        <TITLE>String</TITLE>
        <SALARY>3.14159</SALARY>
        <HIREDATE>1967-08-13</HIREDATE>
        <AREA>String</AREA>
      </ROW>
      <ROW>
        <PIN>String</PIN>
        <LASTNAME>String</LASTNAME>
        <FIRSTNAME>String</FIRSTNAME>
        <MIDINITIAL>String</MIDINITIAL>
        <DIV>String</DIV>
        <DEPT>String</DEPT>
        <JOBCLASS>String</JOBCLASS>
        <TITLE>String</TITLE>
        <SALARY>3.14159</SALARY>
        <HIREDATE>1967-08-13</HIREDATE>
        <AREA>String</AREA>
      </ROW>
      <ROW>
        <PIN>String</PIN>
        <LASTNAME>String</LASTNAME>
        <FIRSTNAME>String</FIRSTNAME>
        <MIDINITIAL>String</MIDINITIAL>
        <DIV>String</DIV>
        <DEPT>String</DEPT>
        <JOBCLASS>String</JOBCLASS>
        <TITLE>String</TITLE>
        <SALARY>3.14159</SALARY>
        <HIREDATE>1967-08-13</HIREDATE>
        <AREA>String</AREA>
      </ROW>
    </RESULTSET_1>
  </paramempdata>
</RESULT>

```

Stored Procedure Schemas for an Oracle Database

The following examples are based on schemas created for a stored procedure for an Oracle database.

Example: Stored Procedure Request Schema for an Oracle Database

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Generated by the iBSE 2004-01-13T22:12:21Z -->
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="RDBMS">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="PROCIN">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="Y" type="xsd:string"/>
            </xsd:sequence>
            <xsd:attribute name="location" type="xsd:string"
use="optional" fixed="RDBMS/Schemas/EDARPK/Procedures/PROCIN"/>
          </xsd:complexType>
        </xsd:element>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

Example: Stored Procedure Request Instance Document for an Oracle Database

```
<?xml version="1.0" encoding="UTF-8"?>
<RDBMS xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="D:\iway\RDBMS\PROCIN_request.xsd">
  <PROCIN location="RDBMS/Schemas/EDARPK/Procedures/PROCIN">
    <Y>String</Y>
  </PROCIN>
</RDBMS>
```

Example: Stored Procedure Response Schema for an Oracle Database

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Generated by the iBSE 2004-01-13T22:18:44Z -->
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="RESULT">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="PROCIN">
          <xsd:complexType>
            <xsd:sequence/>
          </xsd:complexType>
        </xsd:element>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

Example: Stored Procedure Response Instance Document for an Oracle Database

```
<?xml version="1.0" encoding="UTF-8"?>
<RESULT xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="D:\iway\RDBMS\PROCIN_response.xsd">
  <PROCIN/>
</RESULT>
```

Stored Procedure Schemas for a VSAM Database

The following examples are based on schemas created for an iWay stored procedure for a VSAM database.

Example: Stored Procedure Request Schema for a VSAM Database

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Generated by the iBSE 2004-01-13T22:05:56Z -->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">
  <xs:element name="RPCIn">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="1" type="xs:string"/>
      </xs:sequence>
      <xs:attribute name="name" type="xs:string" use="optional"
default="RPCVSM"/>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Example: Stored Procedure Request Instance Document for a VSAM Database

```
<?xml version="1.0" encoding="UTF-8"?>
<RPCIn xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="D:\iway\RDBMS\RPCVSM_request.xsd"
name="RPCVSM">
  <l>String</l>
</RPCIn>
```

Example: Stored Procedure Response Schema for a VSAM Database

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Generated by the iBSE 2004-01-13T22:05:56Z -->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">
  <xs:element name="RPCOut">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Row" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="COMP_NAME" type="xs:string"/>
              <xs:element name="EMP_ID" type="xs:string"/>
              <xs:element name="EMPID" type="xs:string"/>
              <xs:element name="FIRST_NAME" type="xs:string"/>
              <xs:element name="LAST_NAME" type="xs:string"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
      <xs:attribute name="status" type="xs:string" use="required"/>
      <xs:attribute name="reason" type="xs:string" use="required"/>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Example: Stored Procedure Response Instance Document for a VSAM Database

```

<?xml version="1.0" encoding="UTF-8"?>
<RPCOut xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="D:\iway\RDBMS\RPCVSM_response.xsd"
status="String" reason="String">
  <Row>
    <COMP_NAME>String</COMP_NAME>
    <EMP_ID>String</EMP_ID>
    <EMPID>String</EMPID>
    <FIRST_NAME>String</FIRST_NAME>
    <LAST_NAME>String</LAST_NAME>
  </Row>
  <Row>
    <COMP_NAME>String</COMP_NAME>
    <EMP_ID>String</EMP_ID>
    <EMPID>String</EMPID>
    <FIRST_NAME>String</FIRST_NAME>
    <LAST_NAME>String</LAST_NAME>
  </Row>
  <Row>
    <COMP_NAME>String</COMP_NAME>
    <EMP_ID>String</EMP_ID>
    <EMPID>String</EMPID>
    <FIRST_NAME>String</FIRST_NAME>
    <LAST_NAME>String</LAST_NAME>
  </Row>
</RPCOut>

```

Multiple Executions of the Same SQL Statement or Stored Procedure

The iWay XML Adapter for RDBMS allows you to execute a prepared SQL statement or stored procedure multiple times with different input parameters each time. The major benefits of this feature are as follows:

- **Connection Resource Utilization.** One connection or thread is established to the back-end database. This minimizes resource consumption. Note that the number of cursors for select statements created is based on the number of parameter sets in the submitted request. If there are 3 sets of parameters for a select statement, there will be 3 cursors created and closed sequentially.
- **Logical Unit of Work (LUW).** A logical unit of work (LUW) is established that will roll back all of the updates or inserts that were issued by the SQL statement of prior executions. A transaction starts at the first set of parameters. The sets of parameters are executed in order from top to bottom of the message sequentially. For example, if an insert statement is submitted along with 3 sets of parameters, and the third set of parameters fails to insert, the previous 2 inserts will be rolled back.
- **Integration of Data From an Outside Source.** For integration scenarios where external data is used to “feed” SQL or stored procedures, this feature allows external data to be mapped to one XML execution block for the adapter to execute.

Note: Multiple processes in one message is not supported. Only one SQL statement or stored procedure can be executed in a single XML execution request block.

Example: Stored Procedure Request Instance Document for an SQL Server Database

The following is a sample XML request document to execute a Microsoft SQL Server stored procedure called CustOrderHist. Note that the PARAMS tag surrounds the parameter(s) to be used as input. The adapter executes the stored procedure twice, with different input values for each execution.

```
<RDBMS>
  <ITERATE location="RDBMS/Schemas/dbo/Procedures/CustOrderHist">
    <PARAMS>
      <parm>ALFKI</parm>
    </PARAMS>
    <PARAMS>
      <parm>NNN*</parm>
    </PARAMS>
  </ITERATE>
</RDBMS>
```

Example: Stored Procedure Response Instance Document for an SQL Server Database

The following is a sample XML response document from the stored procedure CustOrderHist. Note the multiple result sets that correspond to the two executions of the stored procedure. The second and last result set contains 0 records found for the input value "NNN*".

```
<?xml version="1.0" encoding="UTF-8" ?>
- <RESULTS>
- <RESULT>
- <CustOrderHist>
- <RESULTSET_1>
- <ROW>
  <ProductName>Aniseed Syrup</ProductName>
  <Total>6</Total>
</ROW>
- <ROW>
  <ProductName>Chartreuse verte</ProductName>
  <Total>21</Total>
</ROW>
- <ROW>
  <ProductName>Escargots de Bourgogne</ProductName>
  <Total>40</Total>
</ROW>
- <ROW>
  <ProductName>Flotemysost</ProductName>
  <Total>20</Total>
</ROW>
- <ROW>
  <ProductName>Grandma's Boysenberry Spread</ProductName>
  <Total>16</Total>
</ROW>
- <ROW>
  <ProductName>Lakkalikööri</ProductName>
  <Total>15</Total>
</ROW>
- <ROW>
  <ProductName>Original Frankfurter grüne Soße</ProductName>
  <Total>2</Total>
</ROW>
- <ROW>
  <ProductName>Raclette Courdavault</ProductName>
  <Total>15</Total>
</ROW>
</RESULTSET_1>
<RETURN_VALUE>0</RETURN_VALUE>
</CustOrderHist>
</RESULT>
```

```
- <RESULT>
- <CustOrderHist>
  <RESULTSET_1 />
  <RETURN_VALUE>0</RETURN_VALUE>
</CustOrderHist>
</RESULT>
</RESULTS>
```

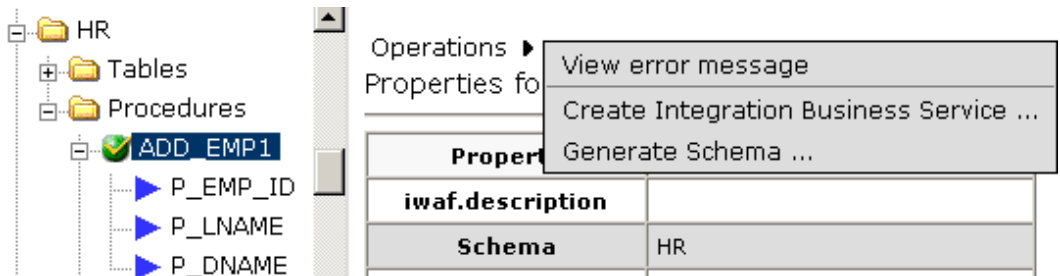
Stored Procedures with Constraints

Certain stored procedures contain constraints, meaning that if no value is supplied for an insert, a default value that meets the constraints is supplied. The adapter does not allow the creation of schemas unless these constraints are met.

The adapter allows you to input default parameters in order to meet the constraints and generate your schemas. You must view and understand an error message before determining if setting parameter values will resolve the issue.

Procedure: How to View an Error Message and Set Parameter Values

The following image shows a stored procedure selected in the left pane and the Operations menu in the right pane.



To view an error message:

1. After you create a schema for a stored procedure with constraints, select the stored procedure in the left pane.
2. In the right pane, move the pointer over *Operations*, and select *View error message*.

If an error message similar to the one in the following image appears, you have the option to Set Parameter Values.

View error message

Error: :

```
ORA-02290: check constraint  
(HR.DNAME_CK) violated  
ORA-06512: at "HR.ADD_EMP1", line 8  
ORA-06512: at line 1
```

Set Parameter Values

Cancel

3. Click *Set Parameter Values*.

The View error message pane opens on the right and contains columns for Parameter Name, Data Type, Column Type, and Default Value.

View error message

Parameter

Parameter Name	Data Type	Column Type	Default Value
P_EMP_ID	NUMBER	IN <input type="button" value="v"/>	<input type="text"/>
P_LNAME	VARCHAR2	IN <input type="button" value="v"/>	<input type="text"/>
P_DNAME	VARCHAR2	IN <input type="button" value="v"/>	ACCOUNTING

4. In the Default Value fields, type all of your default constraints.
In this example, the P_DNAME field is the one with the constraint, so it requires a default value, for example, ACCOUNTING.
5. After you enter the value(s), click *Update* to return to the main properties window.
6. In the right pane, move the pointer over Operations and select *Test Run*.

The Test Run pane opens on the right, where you can test the stored procedure. ACCOUNTING is automatically populated as the default value for P_DNAME.

Test Run

Parameter

Parameter Name	Data Type	Column Type	Value
P_EMP_ID	NUMBER	IN	<input type="text" value="100"/>
P_LNAME	VARCHAR2	IN	<input type="text" value="SMITH"/>
P_DNAME	VARCHAR2	IN	<input type="text" value="ACCOUNTING"/>

7. Enter additional parameter values (for example, 100 and SMITH) and click *Test*.

The results appear in the Test Run results window as shown in the following image.

Test Run

results :

```
<RESULT>
  <ADD_EMP1/>
</RESULT>
```

OK

8. To return to the main properties window where you can generate schemas and create Web services for the stored procedure, click *OK*.

Understanding iWay Business Services

Application Explorer provides Web developers with a simple, consistent mechanism for extending the capabilities of the adapter. The iWay Business Services Engine (iBSE) exposes functionality as Web services. It serves as a gateway to heterogeneous back-end applications and databases.

A Web service is a self-contained, modularized function that can be published and accessed across a network using open standards. It is the implementation of an interface by a component and is an executable entity. For the caller or sender, a Web service can be considered as a “black box” that may require input and delivers a result. A Web service integrates within an enterprise as well as across enterprises on any communication technology stack, whether asynchronous or synchronous, in any format.

Note: In a J2EE Connector Architecture (JCA) implementation of iWay adapters, Web services are not available. When the adapters are deployed to use the iWay Connector for JCA, the Common Client Interface provides integration services using the iWay adapters. For more information, see the *iWay Installation and Configuration* manual and the *iWay Connector for JCA User's Guide*.

Creating a Business Service

You can create a business service for an SQL statement or a stored procedure.

Procedure: How to Generate a Business Service

To generate a business service:

1. If you are not connected to a defined target, connect to one, as described in *How to Connect to a Defined Target* on page 2-8.
2. Expand the node to display the statements or procedures.
3. Click the SQL statement or stored procedure for which you want to create a business service.
4. In the right pane, move the pointer over *Operations* and select *Create iWay Business Services*.

The Create Web Service information appears in the right pane.

5. Choose whether to create a new service or use an existing service.

If you select *Use an existing service*, a drop-down list appears from which you must select the service.

If you select *Create a new service*, the Create Web Service pane opens on the right as shown in the following image.

Create Web Service for Param Car

Service Name:

Description:

License:

- production
- test

Help < Back Next > Cancel

- a. In the Service Name field, type a name to identify the Web service (under the Service node in the left pane of the iWay Business Services tab).
 - b. In the Description field, type a brief description of the Web service.
 - c. In the License field, select the license(s) with which you want to associate this business service. To select more than one, hold down the *Ctrl* key and click the licenses.
- 6.** Click *Next*.

Another pane with the Method Name and Description fields opens.

- a. In the Method Name field, type a name to specify the name of the SQL statement or stored procedure to be added to the business service.
 - b. In the Description field, type a brief description of the method.
- 7.** Click *Finish*.

Application Explorer switches the view to the iWay Business Services tab, and the new business service appears in the left pane.

Testing a Business Service

After a business service is created, test it to ensure that it functions properly. iWay provides a test tool for testing the business service.

Procedure: How to Test a Business Service

To test a business service:

1. If you are not on the iWay Business Services tab of Application Explorer, click the tab to access business services.
2. If it is not expanded, expand the list of business services under iWay Business Services.
3. Expand the *Services* node.
4. Select the name of the business service you want to test.

The business service name appears as a link in the right pane.

5. In the right pane, click the named business services link.

The test option appears in the right pane.

If you are testing a Web service that requires XML input, an input field appears as shown in the following illustration. Options to browse, upload, view additional information, or invoke the input are available through buttons.

Test

To test the operation using the [SOAP protocol](#), click the 'Invoke' button.



The screenshot shows a window titled "input xml:". Inside the window is a large, empty text area with a vertical scrollbar on the right side. Below the text area is a horizontal row of four buttons: "Browse...", "Upload", "More", and "Invoke". The "Browse..." button is highlighted with a yellow background.

If you are testing a Web service for a parameterized SQL statement, an input area appears where you can enter the parameter value, as shown in the following illustration.

Click [here](#) for a complete list of operations.

paramcardata

Test

To test the operation using the [SOAP protocol](#), click the 'Invoke' button.

Parameter	Value
param0:	<input type="text"/>

6. Provide the input for the appropriate input pane.
7. Click *Invoke*.

Application Explorer displays the results in the right pane as shown in the following illustration.

```

<?xml version="1.0" encoding="UTF-8" ?>
- <SOAP-ENV:Envelope
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:SOAP-
  ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-
  instance">
- <SOAP-ENV:Body>
  - <SPmethResponse
    xmlns="urn:iwaysoftware:ibse:jul2003:SPmeth:respo
    cid="16B82D8AAFC70DF1B04C2C80028A05C8">
  - <RPCOut reason="" status="success">
    - <Row>

      <DBLNAME>SYSDATABASE</DBLNAME>
      <EDAACCESS>R</EDAACCESS>
      <ISOLAT>C</ISOLAT>
      <IDENTIFY>N</IDENTIFY>
      <ENGINE>EDA</ENGINE>
      <DBDESCR>EDA Database
      Information Tables</DBDESCR>
    </Row>
  - <Row>

      <DBLNAME>SYSEXTENDED</DBLNAME>
      <EDAACCESS>R</EDAACCESS>
      <ISOLAT>C</ISOLAT>
      <IDENTIFY>N</IDENTIFY>
  
```

Generating WSDL From a Web Service

Generating Web Services Description Language (WSDL) from a Web service enables you to make the Web service available to other services within a host server such as Sun Java System Application Server.

Procedure: How to Generate WSDL From a Web Service

To generate WSDL from a Web service:

1. If you are not already in the iWay Business Services tab, click the tab to access business services.
2. In the left pane, expand the list of services to display the Web service for which you want to generate WSDL.
3. Click the Web service.

The link for the service appears in the right pane.

4. Right-click the *Service Description* link and choose *Save Target As*.
5. Choose a location for the file and specify *.wsdl* for the extension.

Note: The file extension must be *.wsdl*.

6. Click *Save*.

Example: Viewing WSDL Generated from a Web Service

The following is an example of a WSDL file for a Web service called MPS generated from a parameterized SQL statement against a VSAM database.

```
<definitions xmlns:tns="urn:schemas-iwaysoftware-com:iwse"
targetNamespace="urn:schemas-iwaysoftware-com:iwse"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
xmlns:m11="urn:iwaysoftware:ibse:jul2003:VSAM:response"
xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:m1="urn:iwaysoftware:ibse:jul2003:VSAM"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"><types><xs:schema
targetNamespace="urn:schemas-iwaysoftware-com:iwse"
elementFormDefault="qualified"><xs:element
name="ibsinfo"><xs:complexType><xs:sequence><xs:element type="xs:string"
name="service"/><xs:element type="xs:string" name="method"/><xs:element
type="xs:string" name="license"/><xs:element type="xs:string"
minOccurs="0" name="disposition"/><xs:element type="xs:string"
minOccurs="0" name="Username"/><xs:element type="xs:string" minOccurs="0"
name="Password"/><xs:element type="xs:string" minOccurs="0"
name="language"/></xs:sequence></xs:complexType></xs:element>
</xs:schema><xs:schema
targetNamespace="urn:schemas-iwaysoftware-com:iwse"
elementFormDefault="qualified"><xs:element
name="adapterexception"><xs:complexType><xs:sequence><xs:element
type="xs:string"
name="error"/></xs:sequence></xs:complexType></xs:element>
</xs:schema><xs:schema
targetNamespace="urn:iwaysoftware:ibse:jul2003:VSAM"
xmlns:m1="urn:iwaysoftware:ibse:jul2003:VSAM"
elementFormDefault="qualified"><xs:element
name="VSAM"><xs:complexType><xs:sequence><xs:element type="xs:string"
name="emp_id"/></xs:sequence></xs:complexType></xs:element>
</xs:schema><xs:schema
targetNamespace="urn:iwaysoftware:ibse:jul2003:VSAM:response"
xmlns:m11="urn:iwaysoftware:ibse:jul2003:VSAM:response"
elementFormDefault="qualified"><xs:element
name="VSAMResponse"><xs:complexType><xs:sequence><xs:element
name="RESULT"><xs:complexType><xs:sequence><xs:element
name="MPSVSAM"><xs:complexType><xs:sequence><xs:element minOccurs="0"
name="RESULTSET_1"><xs:complexType><xs:sequence><xs:element minOccurs="0"
name="ROW" maxOccurs="unbounded"><xs:complexType><xs:sequence><xs:element
type="xs:string" name="EMP_ID"/><xs:element type="xs:string"
name="FIRST_NAME"/><xs:element type="xs:string"
name="LAST_NAME"/><xs:element type="xs:string" name="DEPT"/><xs:element
type="xs:string"
```

```
name="COMP_NAME"/></xs:sequence></xs:complexType></xs:element></xs:sequence></xs:complexType></xs:element></xs:sequence></xs:complexType></xs:element></xs:sequence></xs:complexType></xs:element></xs:sequence><xs:attribute type="xs:string" use="required" name="cid"/></xs:complexType></xs:element></xs:schema> </types><message name="VSAMIn"><part element="m1:VSAM" name="parameters"/></message><message name="VSAMOut"><part element="m11:VSAMResponse" name="parameters"/> </message><message name="MPSHeader"><part element="tns:ibsinfo" name="header"/> </message><message name="AdapterException"><part element="tns:adapterexception" name="fault"/></message><portType name="MPSSoap"><operation name="VSAM"><documentation/><input message="tns:VSAMIn"/><output message="tns:VSAMOut"/><fault message="tns:AdapterException" name="AdapterExceptionFault"/></operation></portType><binding type="tns:MPSSoap" name="MPSSoap"><soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/><operation name="VSAM"><soap:operation style="document" soapAction="MPS.VSAMRequest@production@@"/><input><soap:body use="literal"/><soap:header part="header" message="tns:MPSHeader" use="literal"/></input><output><soap:body use="literal"/></output><fault name="AdapterExceptionFault"><soap:fault use="literal" name="AdapterExceptionFault"/></fault></operation></binding><service name="MPS"><documentation>MPS</documentation><port binding="tns:MPSSoap" name="MPSSoap1"><soap:address location="http://iwayntk1:7001/ibse/IBSEServlet/XDSOAPRouter"/></port></service></definitions>
```

Identity Propagation

If you test or execute a Web service using a third party XML editor, for example XMLSPY, the Username and Password values that you specify in the SOAP header must be valid and are used to connect to RDBMS. The user name and password values that you provided for RDBMS during target creation using Application Explorer are overwritten for this Web service request.

The following is a sample SOAP header that is included in the WSDL file for a Web service:

```
<SOAP-ENV:Header>
  <m:ibsinfo xmlns:m="urn:schemas-iwaysoftware-com:iwse">
    <m:service>String</m:service>
    <m:method>String</m:method>
    <m:license>String</m:license>
    <m:disposition>String</m:disposition>
    <m:Username>String</m:Username>
    <m>Password>String</m>Password>
    <m:language>String</m:language>
  </m:ibsinfo>
</SOAP-ENV:Header>
```

Note: You can remove the following tags from the SOAP header, since they are not required:

```
<m:disposition>String</m:disposition>
<m:language>String</m:language>
```

CHAPTER 3

Listening for Database Events

Topics:

- Understanding iWay Event Functionality
- Creating, Editing, or Deleting an Event Port
- Creating, Editing, or Deleting an Event Channel
- Choosing a Listening Technique
- Standard Event Processing With Row Tracking
- Standard Event Processing With Row Removal
- Trigger-based Event Processing

This section describes how to use the iWay XML Adapter for RDBMS, deployed to a server such as Sun Java System Application Server, to listen for events in a relational table. Several listening techniques are available, enabling you to choose the technique that best suits your requirements.

Understanding iWay Event Functionality

Events are generated as a result of activity in a database or application system. You can use events to trigger an action in your application. For example, an update to a database can reflect an update to customer information. If your application must perform an action when this happens, your application is a consumer of this event.

After you create a connection to your application system, you can add events using Application Explorer. To create an iWay Event, you must create a port and a channel.

The following is a description of how ports and channels work:

- Port

A port associates a particular business object exposed by an adapter with a particular disposition. A disposition defines the protocol and resulting location of the event data. The port defines the end point of the event consumption. For more information, see *Creating, Editing, or Deleting an Event Port on page 3-2*.

- Channel

A channel represents configured connections to particular instances of back-end or other types of systems. A channel binds one or more event ports to a particular listener managed by an adapter. For more information, see *Creating, Editing, or Deleting an Event Channel on page 3-17*.

Important: When using the adapter in conjunction with the iWay connector for JCA 1.5, there is no need to create event ports to dispose of event data. However, you must create a channel to enable event listening capabilities.

Creating, Editing, or Deleting an Event Port

The following topics describe how to create, edit, or delete an event port using iWay Servlet Application Explorer.

Important: When using the adapter in conjunction with the iWay connector for JCA 1.5, there is no need to create event ports to dispose of event data. However, you must create a channel to enable event listening capabilities.

Creating an Event Port From the iWay Events Tab

The following procedures describe how to create an event port from the iWay Events tab for various dispositions. You can switch between an iBSE and a JCA deployment by using the drop-down menu in the upper right of Application Explorer.

The following dispositions are available when using Application Explorer in conjunction with an iBSE deployment:

- File
- iBSE
- MSMQ
- JMSQ
- SOAP
- HTTP
- MQ Series

Note: The MAIL disposition option will be supported in a future release.

The following dispositions are available when using Application Explorer in conjunction with a JCA connector deployment.

- File
- JMSQ
- HTTP
- MQ Series

Important: When using the adapter in conjunction with the iWay connector for JCA 1.5, there is no need to create event ports to dispose of event data. However, you must create a channel to enable event listening capabilities.

Procedure: How to Create an Event Port for File

To create an event port for File:

1. Click the *iWay Events* tab.
2. In the left pane, expand the *RDBMS* node.
3. Select the *ports* node.
4. In the right pane, move the pointer over *Operations* and select *Add a new port*.

The Create New Port pane opens on the right as shown in the following image.

Create New Port

Choose parameters of the port that you wish to create.

Port Name:

Description:

Disposition Protocol:

Disposition:

- a. In the Name field, type a name for the event port.
- b. In the Description field, type a brief description.
- c. From the Disposition Protocol drop-down list, select *FILE*.
- d. In the Disposition field, specify a destination file to which the event data is written.

When pointing Application Explorer to an **iBSE** deployment, specify the destination file using the following format:

`ifile://[location];errorTo=[pre-defined port name or another disposition url]`

When pointing Application Explorer to a **JCA** deployment, specify the full path to the directory.

Important: When using the adapter in conjunction with the iWay Connector for JCA 1.5, there is no need to create event ports to dispose of event data. However, you must create a channel to enable event listening capabilities.

The following table lists and defines the parameters for the File disposition.

Parameter	Description
location	Destination and file name of the document where event data is written, for example: <code>D:\in\x.txt</code>
errorTo	Location to which error logs are sent. Optional. Predefined port name or another disposition URL. The URL must be complete, including the protocol.

5. Click *OK*.

The port appears under the ports node in the left pane. In the right pane, a table appears that summarizes the information associated with the event port you created.

You are ready to associate the event port with a channel. For more information, see *Creating, Editing, or Deleting an Event Channel* on page 3-17.

Procedure: How to Create an Event Port for iBSE

To create an event port for iBSE:

1. Click the *iWay Events* tab.
2. In the left pane, expand the *RDBMS* node.
3. Select the *ports* node.
4. In the right pane, move the pointer over *Operations* and select *Add a new port*.

The Create New Port pane opens on the right as shown in the following image.

- a. In the Name field, type a name for the event port.
- b. In the Description field, type a brief description.
- c. From the Disposition Protocol drop-down list, select *IBSE*.
- d. In the Disposition field, type an iBSE destination using the following format:

```
ibse: [svcName] . [mthName] ; responseTo= [pre-defined port name or
another disposition url] ; errorTo= [pre-defined port name or another
disposition url]
```

The following table lists and defines the parameters for the iBSE disposition.

Parameter	Description
svcName	Name of the service created with iBSE.
mthName	Name of the method created for the Web service.
responseTo	Location to which responses are posted. Optional. A predefined port name or another disposition URL. The URL must be complete, including the protocol.
errorTo	Location to which error logs are sent. Optional. Predefined port name or another disposition URL. The URL must be complete, including the protocol.

5. Click *OK*.

The port appears under the ports node in the left pane. In the right pane, a table appears that summarizes the information associated with the event port you created.

You are ready to associate the event port with a channel. For more information, see *Creating, Editing, or Deleting an Event Channel* on page 3-17.

Procedure: How to Create an Event Port for MSMQ

To create an event port for MSMQ:

1. Click the *iWay Events* tab.
2. In the left pane, expand the *RDBMS* node.
3. Select the *ports* node.
4. In the right pane, move the pointer over *Operations* and select *Add a new port*.

The Create New Port pane opens on the right as shown in the following image.

Create New Port

Choose parameters of the port that you wish to create.

Port Name:

Description:

Disposition Protocol:

Disposition:

- a. In the Name field, type a name for the event port.
- b. In the Description field, type a brief description.
- c. From the Disposition Protocol drop-down list, select *MSMQ*.
- d. In the Disposition field, type an MSMQ destination using the following format:

```
msmq:/[machineName]/private$/[qName];errorTo=[pre-defined port name or another disposition url]
```

Note: This syntax is for a private queue. Private queues are queues that are not published in Active Directory. They appear only on the local computer that contains them. Private queues are accessible only by Message Queuing applications that recognize the full path name or format name of the queue.

The following table lists and defines the parameters for the MSMQ disposition.

Parameter	Description
machineName	Machine name where the Microsoft Queuing system is running.
qName	Name of the private queue where messages are placed.
errorTo	Location to which error logs are sent. Optional. Predefined port name or another disposition URL. The URL must be complete, including the protocol.

5. Click *OK*.

The port appears under the ports node in the left pane. In the right pane, a table appears that summarizes the information associated with the event port you created.

You are now ready to associate the event port with a channel. For more information, see *Creating, Editing, or Deleting an Event Channel* on page 3-17.

Procedure: How to Create an Event Port for JMSQ

To create an event port for JMSQ:

1. Click the *iWay Events* tab.
2. In the left pane, expand the *RDBMS* node.
3. Select the *ports* node.
4. In the right pane, move the pointer over *Operations* and select *Add a new port*.

The Create New Port pane opens on the right as shown in the following image.

Create New Port

Choose parameters of the port that you wish to create.

Port Name:

Description:

Disposition Protocol:

Disposition:

- a. In the Name field, type a name for the event port.
- b. In the Description field, type a brief description.
- c. From the Disposition Protocol drop-down list, select *JMSQ*.
- d. In the Disposition field, type a JMS destination.

When pointing Application Explorer to an **ibSE** deployment, specify the destination using the following format:

```
jmsq: [myQueueName]@[myQueueFac] ; jndiurl= [myurl] ; jndifactory= [myfactory] ; user= [user] ; password= [xxx] ; errorTo= [pre-defined port name or another disposition url]
```

When pointing Application Explorer to a **JCA** deployment, specify the destination using the following format:

```
jms:jmsqueue@jmsfactory; jndiurl=; jndifactory=;
```

Important: When using the adapter in conjunction with the iWay Connector for JCA 1.5, there is no need to create event ports to dispose of event data. However, you must create a channel to enable event listening capabilities.

The following table lists and defines the parameters for the JMSQ disposition.

Parameter	Description
myQueueName or jmsqueue	JNDI name of a queue to which events are emitted.
myQueueFac or jmsfactory	Resource that contains information about the JMS Server. You must create the connection factory; for example, sampleQCF.
jndiurl	URL to use to contact the JNDI provider. The syntax of this URL depends on the JNDI provider being used. This value corresponds to the standard JNDI property, <code>java.naming.provider.url</code> The URL of the Sun Java System Application Server is <code>iiop://localhost:3700</code> where: <code>3700</code> Is a default port.
jndifactory	Is JNDI context.INITIAL_CONTEXT_FACTORY and is provided by the JNDI service provider. For Sun Java System Application Server, this is <code>com.sun.jndi.cosnaming.CNCtxFactory</code>
user	Valid user name required to access a JMS server.
password	Valid password required to access a JMS server.
errorTo	Location to which error logs are sent. Optional. A predefined port name or another disposition URL. The URL must be complete, including the protocol.

5. Click **OK**.

The port appears under the ports node in the left pane. In the right pane, a table appears that summarizes the information associated with the event port you created.

You are now ready to associate the event port with a channel. For more information, see *Creating, Editing, or Deleting an Event Channel* on page 3-17.

Procedure: How to Create a Port for SOAP

To create a port for SOAP:

1. Click the *iWay Events* tab.
2. In the left pane, expand the *RDBMS* node.
3. Select the *ports* node.
4. In the right pane, move the pointer over *Operations* and select *Add a new port*.

The Create New Port pane opens on the right as shown in the following image.

Create New Port

Choose parameters of the port that you wish to create.

Port Name:

Description:

Disposition Protocol:

Disposition:

- a. In the Name field, type a name for the event port.
- b. In the Description field, type a brief description.
- c. From the Disposition Protocol drop-down list, select *SOAP*.
- d. In the Disposition field, type a SOAP destination using the following format:

```
soap: [wsdl-url];soapaction=[myaction];method=[web service
method];namespace=[name space];responseTo=[pre-defined port name
or another disposition url];errorTo=[pre-defined port name or
another disposition url]
```

The following table lists and defines the parameters for the SOAP disposition.

Parameter	Description
wSDL-url	<p>The URL to the WSDL file that is required to create the SOAP message. For example:</p> <p>http://localhost:7001/ibse/IBSEServlet/test/webservice.ibs?wsdl</p> <p>where:</p> <p>webservice</p> <p>Is the name of the Web service you created using Application Explorer.</p> <p>This value can be found by navigating to the iWay Business Services tab and opening the Service Description link in a new window. The WSDL URL appears in the Address field.</p> <p>You can also open the WSDL file in a third party XML editor (for example, XMLSPY) and view the SOAP request settings to find this value.</p>
soapaction	The method that will be called by the SOAP disposition.
method	Web service method you are using. This value can be found in the WSDL file.
namespace	XML namespace you are using. This value can be found in the WSDL file.
responseTo	<p>Location to which responses are posted. Optional.</p> <p>A predefined port name or another disposition URL. The URL must be complete, including the protocol.</p>
errorTo	<p>Location to which error logs are sent. Optional.</p> <p>A predefined port name or another disposition URL. The URL must be complete, including the protocol.</p>

5. Click *OK*.

The port appears under the ports node in the left pane. In the right pane, a table appears that summarizes the information associated with the port you created.

You are now ready to associate the event port with a channel. For more information, see *Creating, Editing, or Deleting an Event Channel* on page 3-17.

Procedure: How to Create an Event Port for HTTP

To create an event port for HTTP:

1. Click the *iWay Events* tab.
2. In the left pane, expand the *RDBMS* node.
3. Select the *ports* node.
4. In the right pane, move the pointer over *Operations* and select *Add a new port*.

The Create New Port pane opens on the right as shown in the following image.

Create New Port

Choose parameters of the port that you wish to create.

Port Name:

Description:

Disposition Protocol:

Disposition:

- a. In the Name field, type a name for the event port.
- b. In the Description field, type a brief description.
- c. From the Disposition Protocol drop-down list, select *HTTP*.
- d. In the Disposition field, type an HTTP destination.

When pointing Application Explorer to an **iBSE** deployment, specify the destination using the following format:

```
ihhttp://[myurl];responseTo=[pre-defined port name or another disposition url];
```

When pointing Application Explorer to a **JCA** deployment, specify the destination using the following format:

<http://host:port/uri>

Important: When using the adapter in conjunction with the iWay Connector for JCA 1.5, there is no need to create event ports to dispose of event data. However, you must create a channel to enable event listening capabilities.

The following table lists and defines the parameters for the HTTP disposition when using an **iBSE** deployment.

Parameter	Description
myurl	URL target for the post operation, for example, http://myhost:1234/docroot
responseTo	Location to which responses are posted. Optional. Predefined port name or another disposition URL. The URL must be complete, including the protocol.

The following table lists and defines the parameters for the HTTP disposition when using a **JCA** deployment.

Parameter	Description
host:port	Combination of the name of the host on which the server resides and the port on which the server is listening for the post operation.
uri	Universal resource identifier that completes the URL specification.

5. Click *OK*.

The port appears under the ports node in the left pane. In the right pane, a table appears that summarizes the information associated with the event port you created.

You are now ready to associate the event port with a channel. For more information, see *Creating, Editing, or Deleting an Event Channel* on page 3-17.

Procedure: How to Create an Event Port for MQSeries

To create an event port for MQSeries:

1. Click the *iWay Events* tab.
2. In the left pane, expand the *RDBMS* node.
3. Select the *ports* node.
4. In the right pane, move the pointer over *Operations* and select *Add a new port*.

The Create New Port pane opens on the right as shown in the following image.

Create New Port

Choose parameters of the port that you wish to create.

Port Name:

Description:

Disposition Protocol:

Disposition:

- a. In the Name field, type a name for the event port.
- b. In the Description field, type a brief description.
- c. From the Disposition Protocol drop-down list, select *MQSeries*.
- d. In the Disposition field, type an MQSeries destination.

When pointing Application Explorer to an **ibSE** deployment, specify the destination using the following format:

```
mqseries:[qManager]/[qName];host=[hostname];port=[port];channel=[
channelname];errorTo=[pre-defined port name or another
disposition url]
```

When pointing Application Explorer to a **JCA** deployment, specify the destination using the following format:

```
mq:qmanager@resqueue;host=;port=;channel=
```

Important: When using the adapter in conjunction with the iWay Connector for JCA 1.5, there is no need to create event ports to dispose of event data. However, you must create a channel to enable event listening capabilities.

The following table lists and defines the parameters for the MQ Series disposition.

Parameter	Description
qManager	Name of the queue manager to which the server must connect.
qName or respqueue	Name of the queue where messages are placed.
host	Host on which the MQ server is located (for the MQ Client only).
port	Number to connect to an MQ server queue manager (for the MQ client only).
channel	Case-sensitive name of the channel that connects with the remote MQ server queue manager (for the MQ client only). SYSTEM.DEF.SVRCONN is the default channel name for MQSeries.
errorTo	Location to which error logs are sent. Optional. Predefined port name or another disposition URL. The URL must be complete, including the protocol.

5. Click *OK*.

The port appears under the ports node in the left pane. In the right pane, a table appears that summarizes the information associated with the event port you created.

You are now ready to associate the event port with a channel. For more information, see *Creating, Editing, or Deleting an Event Channel* on page 3-17.

Editing and Deleting an Event Port

The following procedures describe how to edit and delete an event port.

Procedure: How to Edit an Event Port

To edit an event port:

1. In the left pane, select the event port you want to edit.
2. In the right pane, move the pointer over *Operations* and select *Edit*.

The Edit Port dialog box opens.

3. Make the required changes and click *OK*.

Procedure: How to Delete an Event Port

To delete an event port:

1. In the left pane, select the event port you want to delete.
2. In the right pane, move the pointer over *Operations* and select *Delete*.

A confirmation dialog box opens.

3. To delete the event port you selected, click *OK*.

The event port disappears from the list in the left pane.

Creating, Editing, or Deleting an Event Channel

The following topics describe how to create, edit, or delete a channel for your iWay Event. All defined event ports must be associated with a channel.

Creating a Channel

The following procedure describes how to create a channel using iWay Servlet Application Explorer.

Procedure: How to Create a Channel

To create a channel:

1. Click the *iWay Events* tab.

The adapters that appear in the left pane support events.

2. In the left pane, expand the *RDBMS* node.

The ports and channels nodes appear in the left pane.

3. Click the *channels* node.

4. In the right pane, move the pointer over *Operations* and select *Add a new channel*. The Add a new RDBMS channel pane opens on the right as shown in the following image.

Add a new RDBMS channel

Choose a name and description for the new channel that you wish to create.

Channel Name:

Description:

Channel Type:

- a. In the Channel Name field, type a name, for example, NewChannel.
 - b. In the Description field, type a brief description.
 - c. From the Channel Type drop-down list, select a channel type.
5. Click *Next*.

The Edit channels pane opens on the right with four tabs representing listener parameters as shown in the following image.

Edit channels

[JDBC-ODBC Bridge Parameters](#)
[Oracle Parameters](#)
[SQL Server Parameters](#)
[EDA Server Parameters](#)

Host:

Port:

Database Name:

User:

Password:

Polling Interval:

SQL Query:

Post Query:

Delete Keys:

- a. Select either an Oracle, SQL Server, EDA Server, or JDBC-ODBC Bridge Listener by clicking the appropriate tab.

Note: If you are configuring listening capabilities for a non-relational database, select the EDA Server Listener.

The following table lists and describes the parameters for all of the listeners.

Parameter	Description
Host	Name or URL of the machine where the database is installed.
Port	Port on which the Host database is listening.

Parameter	Description
<p>Database Name</p> <ul style="list-style-type: none"> • For SQL Server and EDA Server Listener <p>SID</p> <ul style="list-style-type: none"> • For Oracle Listener <p>Data Source</p> <ul style="list-style-type: none"> • For JDBC-ODBC Bridge Listener 	<ul style="list-style-type: none"> • Database name of the database where the table specified in the SQL statement is located. <p>Note: When you access a non-relational database, and the server component is an SSCTL server component, the database name must be the service name, and you must specify it. If the server component is installed on USS, you can leave the database field blank. For more information about the server component, see Chapter 1, <i>Introducing the iWay XML Adapter for RDBMS</i>.</p> <ul style="list-style-type: none"> • For an Oracle Listener, the SID is a unique name for the database service, chosen by the database administrator or the person who installed Oracle E-Business Suite. <ul style="list-style-type: none"> • For JDBC-ODBC Bridge Listener, this is the name of the data source configured under the ODBC Driver Manager. For more information, see your ODBC Driver Manager documentation.
User	Database user ID to access the table.
Password	Database password associated with the user ID.
Polling Interval	Interval, in milliseconds, at which to check for new input.

Parameter	Description
SQL Query	<p data-bbox="578 258 1299 288">SQL SELECT statement that the listener issues to poll the table.</p> <p data-bbox="578 310 1250 471">If the SQL statement includes a date column or long text column, you must provide a value for the SQL Post-query parameter. The value you provide must not contain a date column or a long text column. This applies whether you provide an SQL statement here or rely upon the default.</p> <p data-bbox="578 492 1240 551">For example, the following SELECT statement retrieves all unprocessed records from the DISCRETE_JOBS table:</p> <pre data-bbox="578 575 1135 688">SELECT * FROM WIP_DISCRETE_JOBS D WHERE DJ.WIP_ENTITY_ID > (SELECT WIP_ENTITY_ID FROM WIP.TEMP_NEW_WORK_ORDER_ENTITY_ID)</pre> <p data-bbox="578 709 1293 802">Important: When a SQL Query joins two or more tables, a SQL Post Query must be used. Also, do not use a semicolon at the end of a SQL statement for a SQL Query or a SQL Post Query.</p>

Parameter	Description
Post Query	<p>A SQL statement that is executed after each new record is read from the table. Case sensitive: the case used to specify the column names must match the case used in the SELECT statement that polled the table. If the SQL Query property was omitted so that a default SELECT statement polled the table, the case used to specify the column names must match the case used to define the columns in the DBMS native schema.</p> <p>If you do not specify a value for SQL Post-query, each record read from the table is deleted after it is read. How this happens depends on whether you specify the Delete Keys property. If you:</p> <p>Specify the Delete Keys property, by default the adapter issues a DELETE statement with a WHERE clause containing every key column specified for the Delete Keys property.</p> <p>At run time this is faster than if you had not specified the Delete Keys property if there is an index on the key or if there are fewer key columns than there are columns in the SELECT statement that polled the table.</p> <p>Do not specify the Delete Keys property, by default the adapter issues a DELETE statement with a WHERE clause that specifies every column from the SELECT statement that polled the table.</p> <p>You can choose to retain the table data after it is read by specifying a value for this parameter, as shown in the examples that follow.</p> <p>Note: The SQL Post-query and Delete Keys parameters are mutually exclusive, because Delete Keys applies to the default DELETE statement, and SQL Post-query overrides the default DELETE statement. You can provide a value for one or the other, but not for both.</p> <p>There are two field operators, ? and ^, that you can use in a post-query SQL statement. For more information, see <i>The Post-query Parameter Operators</i> on page 3-30.</p> <p>Important: When a SQL Query joins two or more tables, a SQL Post Query must be used. Also, do not use a semicolon at the end of a SQL statement for a SQL Query or a SQL Post Query.</p>

Parameter	Description
Delete Keys	<p>Comma-separated list of key columns to be used in the default DELETE statement. DELETE operates on keys, so specify the table key columns.</p> <p>This is case sensitive: the case used to specify the column names must match the case used in the SELECT statement that polled the table. If the SQL Query property was omitted so that a default SELECT statement polled the table, the case used to specify the column names must match the case used to define the columns in the DBMS native schema.</p> <p>Note: The Delete Keys and SQL Post Query parameters are mutually exclusive, because Delete Keys applies to the default DELETE statement, and SQL Post Query overrides the default DELETE statement. You can provide a value for one or the other, but not for both. For more information, see the description of the SQL Post-query parameter in this table.</p>

- b. Type the system information that is specific to the database on which you are listening based on the descriptions in the previous table.
6. Click *Next*.

The Select Ports pane opens on the right with buttons to enable you to move ports from one area to the other as shown in the following image.

Select Ports

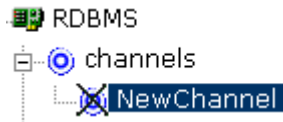
Available		Current	
rdbms1 rdbms2 rdbms3	<div style="margin-bottom: 5px;"><input type="button" value="⏪"/></div> <div style="margin-bottom: 5px;"><input type="button" value="⏴"/></div> <div style="margin-bottom: 5px;"><input type="button" value="⏵"/></div> <div style="margin-bottom: 5px;"><input type="button" value="⏩"/></div>		
<input type="button" value="Help"/>	<input type="button" value="⏴ Back"/>	<input type="button" value="Finish"/>	<input type="button" value="Cancel"/>

- a. Select an event port from the list of available ports. To select more than one, hold down the *Ctrl* key and click the ports.
 - b. Click the single right arrow button to transfer the selected port(s) to the list of current ports. To transfer all event ports, click the double right arrow button.
7. Click *Finish*.

Summary information appears as shown in the following image and includes the channel description, channel status, and current ports. All the information is associated with the channel you created in the right pane.

Operations ▶	
Channel Description	NewChannel
Channel Status	Disconnected
Ports	[rdbms1, rdbms2]

The channel appears under the channels node in the left pane with an X over the icon indicating that the channel is currently disconnected as shown in the following image.



You must start the channel to activate your event configuration.

Procedure: How to Start or Stop a Channel

To start or stop a channel:

1. Expand the *iWay Events* node.
2. Expand the *RDBMS* node.
3. Select the channel you want to start or stop.
4. To start the channel, move the pointer over *Operations* and select *Start the channel*.

The X over the icon disappears as shown in the following image, and the channel starts.



5. To stop the channel, move the pointer over *Operations* and select *Stop the channel*.

Editing or Deleting a Channel

The following procedures describe how to edit or delete a channel.

Procedure: How to Edit a Channel

To edit a channel:

1. Expand the *iWay Events* node.
2. Expand the *RDBMS* node.
3. In the left pane, select the channel you want to edit.
4. In the right pane, move the pointer over *Operations* and select *Edit*.
The Edit channels dialog box opens.
5. Make the required changes to the channel configuration and click *Finish*.

Procedure: How to Delete a Channel

To delete a channel:

1. Expand the *iWay Events* node.
2. Expand the *RDBMS* node.
3. In the left pane, select the channel you want to delete.
4. In the right pane, move the pointer over *Operations* and select *Delete*.
A confirmation dialog box opens.
5. To delete the channel you selected, click *OK*.
The channel disappears from the list in the left pane.

Choosing a Listening Technique

You can detect an event in a relational or non-relational table and propagate it to other processes using a Table Listener.

An elaborate polling technology enables the specification of SQL SELECT statements to execute on a periodic basis. After data is polled, it passes through the adapter for further processing.

Note: Event processing may be limited for some non-relational databases due to the functionality of the database and its interaction with the iWay server component. For more information on the iWay server component, see the *iWay Data Adapter Administrator User's Guide* or consult with your DBA.

You can poll a relational or non-relational database directly and send the results to a file or JMS message queue. You also can use the following advanced techniques to listen to a database event.

- Standard event processing with row tracking

The listener polls a table, sends each newly inserted row to a destination you specify (known as the disposition), and uses a control table to track the row that was most recently read. The control table prevents the most recently read row from being read again during the next listening cycle.

You can apply this flexible yet simple technique in most situations.

For more information, see *Standard Event Processing With Row Tracking* on page 3-27.

- Standard event processing with row removal

The listener polls a table, sends each newly inserted row to a destination you specify, and then deletes the new row from the table to prevent it from being read again during the next listening cycle.

You apply this technique when the source table is used to pass data to the adapter, and the table rows are not required to persist. Rows are deleted as they are processed.

For more information, see *Standard Event Processing With Row Removal* on page 3-33.

- Trigger-based event processing

At design time, you assign triggers to a joined group of tables. At run time, the triggers write information about table changes to a common control table. The listener polls the control table and sends information about the table changes to a destination you specify. The listener deletes new rows from the control table to prevent them from being read again during the next listening cycle.

You apply this technique when listening for events in a group of large joined tables, or when you must know whether a row was updated or deleted.

For more information, see *Trigger-based Event Processing* on page 3-36.

Standard Event Processing With Row Tracking

The standard event processing with row tracking technique enables you to listen to the source table without removing its rows. It requires you to create a single-cell control table that tracks the last new row the Table Listener read from the source table.

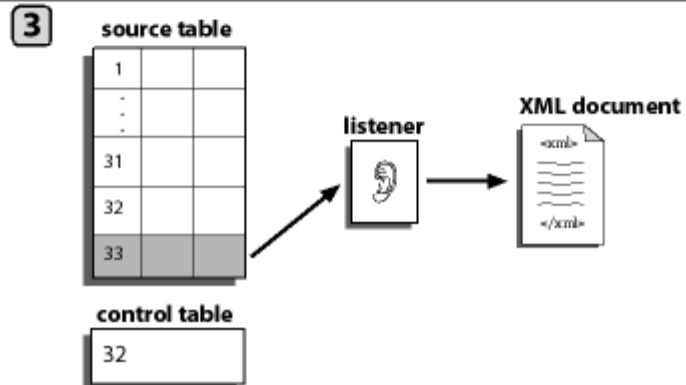
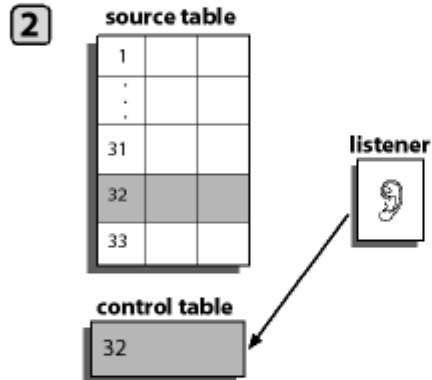
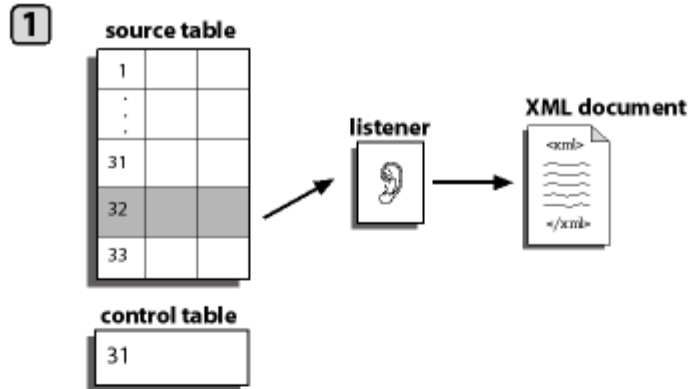
The single column of the control table corresponds to a column (or to a group of columns) in the source table that is unique, sortable, and indicates how recently the row was added to the source table relative to the other rows. For example, the first row added to the source table has the lowest value, and the last row added has the highest value. This value is called the *event key*.

When you create the control table, initialize it to the event key of the row most recently added to the source table. When you specify the listener properties, configure the SQL Post-query property of the listener to automatically update the control table event key.

Each time the listener queries the source table, it looks for rows added since the last query—that is, for rows whose event key is greater than the current value of the field in the control table. It reads each row of this type and returns it to the specific destination using an XML document. To ensure that the row is not read again the next time the listener queries the table, the listener updates the field in the control table to match the value of the row that was just read from the source table.

Note: Event processing may be limited for some non-relational databases due to the functionality of the database and its interaction with the iWay server component. For more information on the iWay server component, see the *iWay Data Adapter Administrator User's Guide* or consult with your DBA.

The following figure illustrates standard event processing with row tracking.



In the previous figure:

1. The listener queries the source table and copies each source table row whose event key is greater than the control table event key. The listener copies the row to an XML document and sends it to the destination defined in the port disposition using the File protocol.
2. The listener updates the event key in the control table to match the row it most recently read.
3. The listener copies the next source table row to an XML document.

The process repeats.

Procedure: How to Implement Standard Event Processing With Row Tracking

To implement standard event processing with row tracking:

1. Create a control table. For an example, see *Creating the Control Table for an RDBMS (Oracle) Event* on page 3-29.
2. Configure an RDBMS Table Listener in the iWay Web Console.

In addition to the required listener properties for standard event processing with row tracking, you also must provide values for the following optional properties:

- **SQL Query**, the SQL SELECT statement that identifies the source table to which the adapter listens and with which it queries the table.
- **SQL Post-query**, the SQL statements that maintain the field in the control table.

For detailed instructions about configuring a listener, see *How to Create a Channel* on page 3-17. For information on post query parameters, see *The Post-query Parameter Operators* on page 3-30.

Example: Creating the Control Table for an RDBMS (Oracle) Event

This example uses an Oracle E-Business Suite (also known as Oracle Applications) table. You can apply the same technique in a similar way to other types of relational databases.

You can follow the steps in this example to create an Oracle E-Business Suite table named TEMP_NEW_YORK_ORDER_ENTITY that has a single field named WIP_ENTITY_ID. You specify this table when you configure the RDBMS Table Listener, as described in *The Post-query Parameter Operators* on page 3-30.

When discrete jobs are created through the Oracle E-Business Suite graphical interface, an entry is created in the WIP.WIP_DISCRETE_JOBS table. For this example, you configure an event to detect new entries to this table. You use the standard event processing with row tracking technique. (Oracle E-Business Suite processing cannot delete rows from the table.)

You first create a simple table to track the records processed.

1. From within Oracle SQL*PLUS, run the following SQL:

```
CREATE TABLE WIP.TEMP_NEW_WORK_ORDER_ENTITY_ID
(
  WIP_ENTITY_ID NUMBER
)
```

This creates a single table with a single field.

Note: Oracle SQL*Plus is part of the Oracle client software. If it is not installed, contact your Oracle Database Administrator.

You must be logged in under the APPS schema or a similar ID with access rights to the Oracle E-Business Suite WIP schema.

2. Create a single record in the table and provide it with the highest WIP_ENTITY_ID ID from your system.

You can obtain this ID from the WIP.WIP_DISCRETE_JOBS table.

This sets the value at which to start detecting events as records enter the WIP_DISCRETE_JOBS table.

3. After you create a simple table in Oracle, you must configure the listener.

Reference: The Post-query Parameter Operators

When you configure a Table Listener, you can use two special field operators, ? and ^, with the SQL Post-query parameter. Both of these operators dynamically substitute database values in the SQL post-query statement at run time:

- ?fieldname is evaluated at run time as field = value.

The ? operator is useful in UPDATE statements:

```
UPDATE table WHERE ?field
```

For example, the following statement

```
UPDATE Stock_Prices_Temp WHERE ?RIC
```

might be evaluated at run time as:

```
UPDATE Stock_Prices_Temp WHERE RIC = 'PG'
```

- ^fieldname is evaluated at run time as value

The ^ operator is useful in INSERT statements:

```
INSERT INTO table VALUES (^field1, ^field2, ^field3, ... )
```

For example, the following statement

```
INSERT INTO Stock_Prices_Temp VALUES (^RIC, ^Price, ^Updated)
```

might be evaluated at run time as:

```
INSERT INTO Stock_Prices_Temp VALUES ('PG', 88.62, '2003-03-18
16:24:00.0')
```

Example: Listening to trans_event Using the Row Tracking Technique

In this example, you listen to the trans_event table using the row tracking technique and use last_trans as the control table that contains the last value of the primary key read from trans_event.

For more information on configuring a listener, see *How to Create a Channel* on page 3-17.

last_trans is to contain a single value in a single row and must be set up prior to configuring the Table Listener. The last_trans column must have the same name as the primary key in the trans_event table. This key must be unique and sortable.

The table schemas for this example are:

```
SQL> describe trans_event
```

Name	Null?	Type
EVENT_ID	NOT NULL	NUMBER(38)
LAST_NAME		VARCHAR2(50)
TRANS_ID		CHAR(2)

```
SQL> describe last_trans
```

Name	Null?	Type
EVENT_ID		NUMBER

The last_trans single field value must contain the starting value of the primary key.

The listener generates XML response documents for each record found in the trans_event table with a primary key greater than the value found in the last_trans table.

1. Using a SQL query/data manipulation tool supplied by the database vendor, insert a record into the trans_event table based on the following information.

- EVENT_ID=1
- LAST_NAME='Kaplan'
- TRANS_ID='03'

When setting up the port, a specific path is configured for a disposition using the File protocol. A response document with the record data is deposited into the directory after the insert is made.

The following is an example of a response document for the listener deposited into a directory specified when the Port is configured.

```
<Oracle>
  <row>
    <EVENT_ID>1</EVENT_ID>
    <LAST_NAME>Kaplan</LAST_NAME>
    <TRANS_ID>03</TRANS_ID>
  </row>
</Oracle>
```

2. Configure the listener by specifying the properties in the following table when creating the channel.

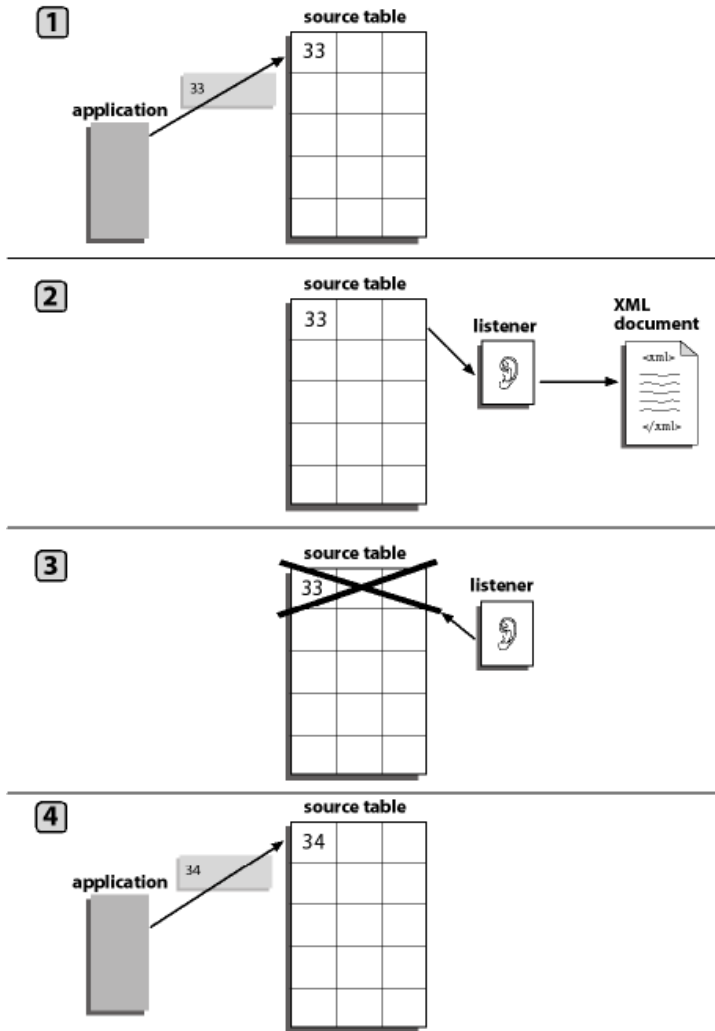
Parameter	Description
Host	Name or URL of the machine on which the database is installed.
Port	Port on which the Host database is listening.
User Name	User name that is registered with the back-end RDBMS.
Password	Password associated with the user name.
SQL Query	<code>SELECT * FROM TRANS_EVENT WHERE EVENT_ID>(select EVENT_ID from LAST_TRANS)</code>
Post Query	<code>UPDATE LAST_TRANS SET ?EVENT_ID</code>
Polling Interval	Interval in seconds.

Standard Event Processing With Row Removal

The standard event processing with row removal technique assumes that the source table is used to pass the data to the adapter and that the table rows are not required to persist. The Table Listener periodically queries the source table. When it finds a row, it reads it and returns it to the file disposition specified when the port is configured via an XML document. To ensure that the row is not read again when the listener next queries the table, the listener deletes the row from the table.

Note: Event processing may be limited for some non-relational databases due to the functionality of the database and its interaction with the iWay server component. For more information on the iWay server component, see the *iWay Data Adapter Administrator User's Guide* or consult with your DBA.

The following figure illustrates standard event processing with row removal.



1. Your application inserts a new row into the source table.
2. The listener queries the source table and copies the new row to an XML document and sends it to the destination defined in the port disposition using the File protocol.
3. The listener deletes the source table row to ensure that the row is not read again when the listener next queries the table.
4. The application inserts a new row into the source table.

The process repeats itself.

Procedure: How to Implement Standard Event Processing With Row Removal

To implement the standard event processing with row removal technique:

1. Configure a Table Listener.
2. In addition to the required listener properties, provide values for the following optional properties:

SQL Query: the SQL SELECT statement that identifies the source table to which the adapter listens and with which it queries the table.

Post Query: the query that identifies the rows that the adapter automatically deletes from the table.

For detailed instructions about configuring a listener, see *How to Create a Channel* on page 3-17. For information on Post query parameters, see *The Post-query Parameter Operators* on page 3-30.

Example: Listening to stock_prices Using the Row Removal Technique

In this example, you listen to the stock_prices table using the row removal technique.

```
SQL> describe stock_prices
```

Name	Null?	Type
RIC	NOT NULL	VARCHAR2 (6)
PRICE		NUMBER (7, 2)
UPDATED		DATE

When a record is added to stock_prices, an XML document is generated with the contents of the record.

The location to which the document is saved is specified in the configuration of the port disposition property (using the File protocol) associated with this Table Listener.

After generating the document, the record is deleted from the table.

1. Configure the listener by specifying the following properties when creating the channel.
 - a. In the Host field, provide the name or URL of the machine on which the database is installed.
 - b. In the Port field, provide the name of the port on which the Host database is listening.
 - c. In the User Name field, provide the user name that is registered with the back-end RDBMS.

- d. In the Password field, provide the Oracle Applications user ID authorized to access the Oracle Applications system.
- e. For the SQL Query, use `select * from stock_prices`.
- f. For the Post Query, use `delete from stock_price where ?RIC`.
- g. For Polling Interval, specify an interval in seconds.

For a description of these properties, see *The Post-query Parameter Operators* on page 3-30.

- 2. For more information on configuring a listener, see *How to Create a Channel* on page 3-17.

Trigger-based Event Processing

Trigger-based event processing is a technique for listening to multiple joined relational tables. You also can use it to detect when a row was deleted or updated.

The trigger-based technique provides the following benefits:

- Improves performance when listening for events in a group of large joined tables.

When processing joined tables, the database system creates a Cartesian product working table. When the joined tables are large, the interim working table is very large. The standard technique of processing database events, in which the adapter periodically listens to the entire structure of joined tables, can consume a significant amount of computing resources.

The trigger-based technique avoids this overhead by requiring the Table Listener to query a single small control table and by writing to the control table only when an event actually occurs.

- Increases the number of event types that the adapter recognizes.

Using the trigger-based technique, you can tell when a row was updated, deleted, or inserted. Using the standard technique, you can tell only when a row was inserted.

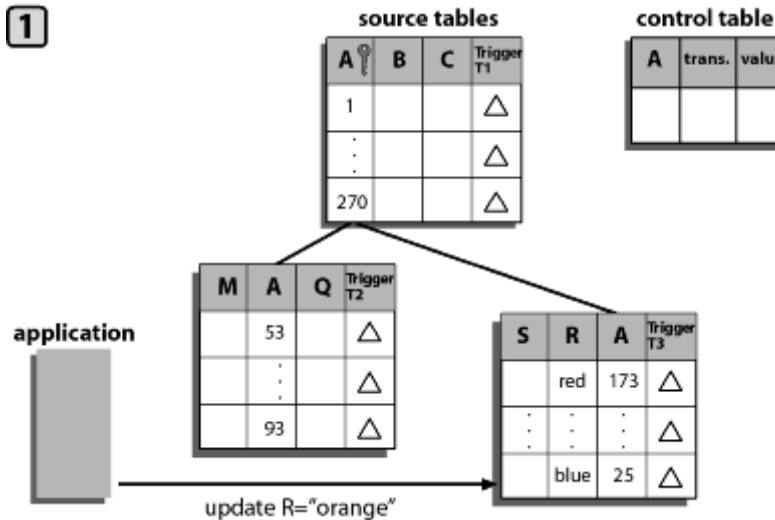
To use the trigger-based technique, you assign a trigger to each table that you want to monitor. When a value changes, it fires the corresponding trigger that writes data to a control table. The iWay XML Adapter for RDBMS listens to the control table by running a query against it. When it finds a row in the control table, it reads it and returns it to the port disposition created when the port is configured via an XML document. To ensure the row is not read again when the listener next queries the table, the listener deletes the row from the table.

The trigger-based technique enables you to recognize changes to an entity. For the purposes of this discussion, an entity is a real-world object that is represented in the database by a hierarchical set of tables.

You manage the triggers using a native RDBMS tool (such as SQL*Plus for Oracle tables) and configure the listener using the iWay Web Console.

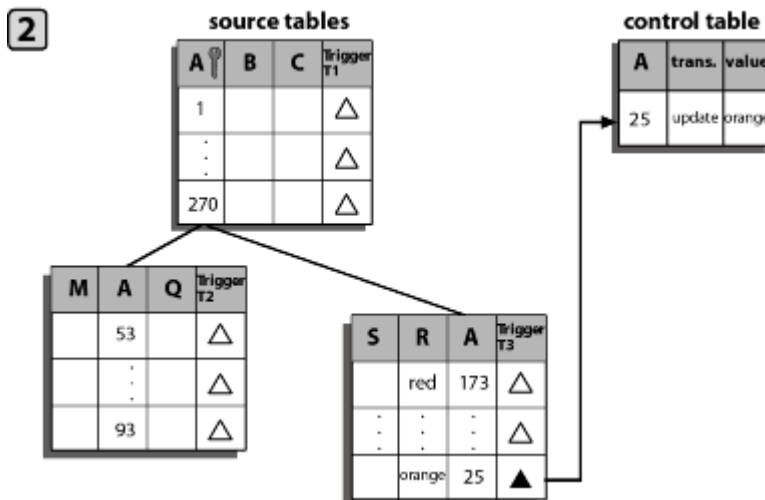
Note: Event processing may be limited for some non-relational databases due to the functionality of the database and its interaction with the iWay server component. For more information on the iWay server component, see the *iWay Data Adapter Administrator User's Guide* or consult with your DBA.

The following figures illustrate trigger-based event processing:



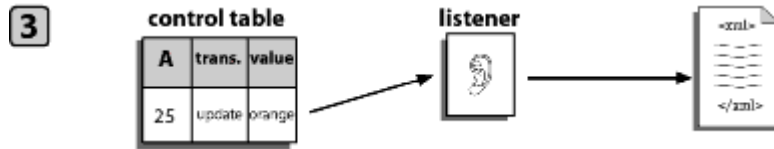
1. Your application updates a row in a group of related source tables as shown in the previous figure.

The update causes a row trigger to fire in the changed table as shown in the following figure.

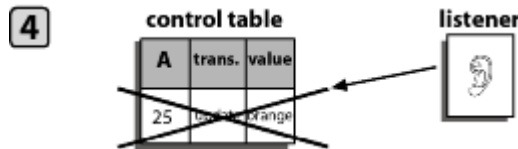


2. The trigger inserts a row into the control table, and the new control table row includes the key value (25), the type of transaction (update), and the new cell value (orange) as shown in the previous figure.

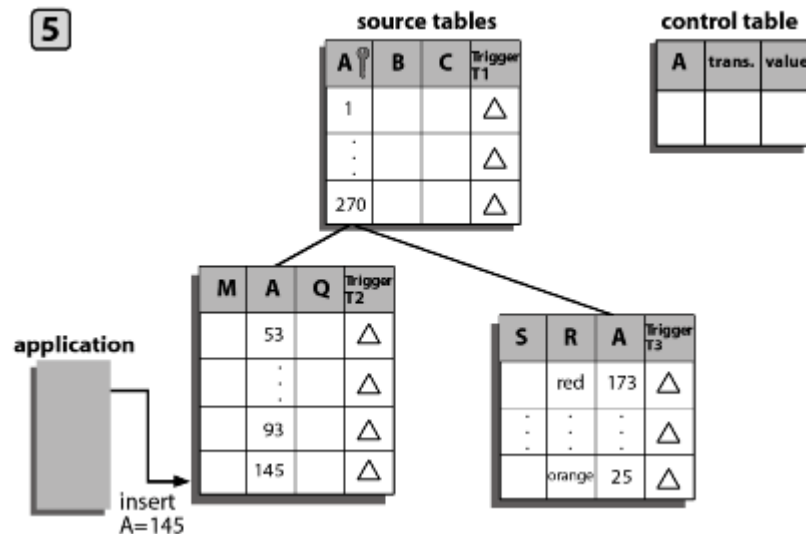
- The listener queries the control table and copies the new row to an XML document. It sends the document to the Reply_to destination as shown in the following figure.



- The listener deletes the control table row to ensure that the row is not read again when the listener next queries the table as shown in the following figure.



- The application inserts a new row into one of the source tables as shown in the following figure.



The process repeats itself.

Procedure: How to Implement Trigger-based Event Processing

To implement the trigger-based event processing technique:

- Create the control table.

The purpose of the control table is to capture the key of each entity that changed, regardless of which entity table changed.

You can store a variety of information in the control table, including the key of the entity that was inserted, updated, or deleted and the name of the table and field that was updated.

The design of the control table is a function of the business logic of your application. For example, you can choose between creating one control table for a group of joined source tables or one control table per source table. Among the issues to consider are the kinds of events to monitor (insertions, deletions, or updates), and whether you want to monitor only the highest-level table in a group of joined tables or all of the tables in the group.

2. Assign triggers to the source tables.

The triggers you assign, and to which tables you assign them, is determined by what kind of change you want to monitor. The triggers implement event-processing logic. For a sample trigger, see *Trigger on WIP_ENTITY_NAME Column in an Oracle Table on page 3-41*.

For example, consider a bill of materials scenario. (A bill of materials is a list of all the parts required to manufacture an item, the subparts required for the parts, and so on. The complete item/parts/subparts relationship can extend to several levels, creating a data structure like a tree with the finished item as the root.) In a bill of materials, each level in the parts hierarchy is represented by a separate table. You might assign a trigger to only the highest-level table (the finished product), or you might assign triggers to all tables (the finished product and its parts and subparts).

If multiple changes are made to the same row during one listener cycle, you could configure the event adapter to record all the changes. If a row was inserted and then updated, both changes are logged.

3. Configure the listener when creating a channel in the Application Explorer console.

In addition to the required listener properties, for trigger-based event processing you also must provide values for the following optional properties:

SQL Query: the SQL SELECT statement that identifies the control table to which the adapter listens and with which it queries the table to determine changes in the source tables.

Post Query: the query that identifies the rows that the adapter automatically deletes from the control table.

For detailed instructions about configuring a listener, see *How to Create a Channel on page 3-17*. For information about Post query operators, see *The Post-query Parameter Operators on page 3-30*.

Example: Trigger on WIP_ENTITY_NAME Column in an Oracle Table

The following trigger fires when a change is made to the WIP_ENTITY_NAME column of the WIP.WIP_ENTITIES Oracle E-Business Suite table. When it fires, the trigger writes the relevant values to the control table IWAY.IWAY_PO_CDC.

```
CREATE OR REPLACE TRIGGER IWAY.IWAY_PO_CDC_WE_TRG

AFTER INSERT OR DELETE OR UPDATE OF WIP_ENTITY_NAME
ON WIP.WIP_ENTITIES
FOR EACH ROW
BEGIN
  IF INSERTING THEN
    INSERT INTO IWAY.IWAY_PO_CDC
      VALUES (
        :NEW.WIP_ENTITY_ID,
        :NEW.ORGANIZATION_ID,
        'UPDATE' );

  ELSE
    INSERT INTO IWAY.IWAY_PO_CDC
      VALUES (
        :OLD.WIP_ENTITY_ID,
        :OLD.ORGANIZATION_ID,
        'UPDATE' );

  END IF;

EXCEPTION
  WHEN DUP_VAL_ON_INDEX THEN
    NULL;          -- Record already exists

END;
```

CHAPTER 4

Using Web Services Policy-Based Security

Topics:

- iWay Business Services Policy-Based Security
- Configuring iWay Business Services Policy-Based Security

Servlet Application Explorer provides a security feature called iWay Business Services policy-based security. The following topics describe how this feature works and how to configure it.

iWay Business Services Policy-Based Security

iWay Business Services provide a layer of abstraction between the back-end business logic they invoke and the user or application running the business service. This enables easy application integration but raises the issue of controlling the use and execution of critical and sensitive business logic that is run as a business service.

Servlet Application Explorer controls the use of business services that use adapters with a feature called policy-based security. This feature enables an administrator to apply *policies* to iWay Business Services (iBS) to deny or permit their execution.

A *policy* is a set of privileges associated with the execution of a business service that can be applied to an existing or new iBS. When you assign specific rights or privileges inside a policy, you need not recreate privileges for every iBS that has security issues in common with other iWay Business Services. Instead, you can use one policy for many iWay Business Services.

The goal is to secure requests at both the transport and the SOAP request level that is transmitted on the wire. Some policies do not deal with security issues directly but affect the run-time behavior of the business services to which they are applied.

The iBSE administrator creates an instance of a policy type, names it, associates individual users and/or groups (a collection of users), and then applies the policy to one or more business services.

You can assign a policy to an iBS or to a method within an iBS. If a policy is applied only to a method, other methods in that iBS are not governed by it. However, if a policy is applied to the iBS, all methods are governed by it. At run time, the user ID and password that are sent to iBSE in the SOAP request message are checked against the list of users for all policies applied to the specific iBS. The Resource Execution policy type is supported and dictates who can or cannot execute the iBS.

When a policy is not applied, the default value for an iBS is to “grant all.” For example, anyone can execute the iBS until the Resource Execution policy is associated to the iBS. At that time, only users granted execution permission, or those who do not belong to a group that was denied execution permissions, have access to the iBS.

Configuring iWay Business Services Policy-Based Security

Before you create instances of policies, you must have a minimum of one user or one group to associate to an instance. You can create users and groups using Servlet Application Explorer. For more information, see *How to Create a User to Associate With a Policy* on page 4-3 or *How to Create a Group to Associate With a Policy* on page 4-5.

An execution policy governs who can execute the business service to which the policy is applied. For more information, see *How to Create an Execution Policy* on page 4-7.

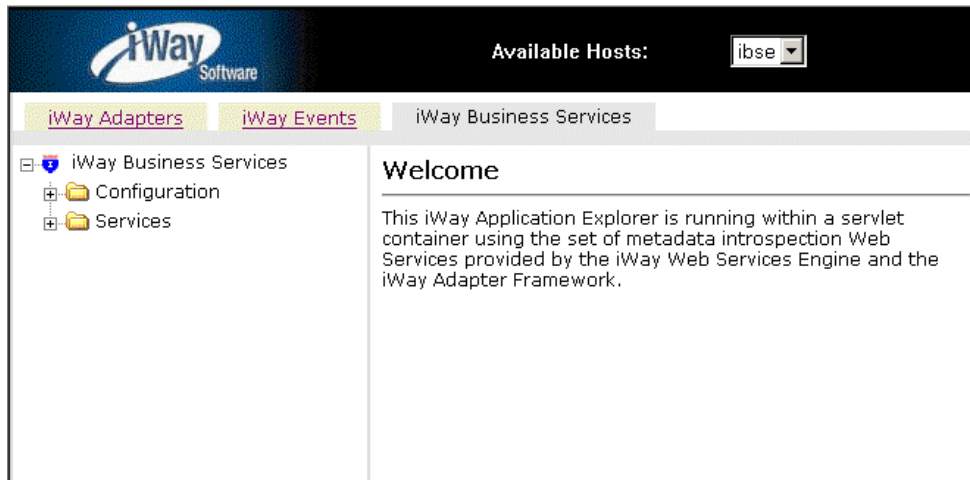
You configure the IP and Domain Restriction policy type slightly differently from other policy types. The IP and Domain Restriction policy type controls connection access to iBSE and therefore, need not be applied to an individual business service. You need not create a policy, however, you must enable the Security Policy option in Servlet Application Explorer. For more information, see *How to Configure IP and Domain Restrictions* on page 4-10.

Procedure: How to Create a User to Associate With a Policy

To create a user to associate with a policy:

1. Open Servlet Application Explorer.

The following image shows the window that opens and includes three tabs corresponding to iWay Adapters, iWay Events, and iWay Business Services. The iWay Business Services tab is active and displays a Welcome screen on the right. The image shows the iWay Business Services node expanded in the left pane.



- a. Click the *iWay Business Services* tab.
- b. Expand the *Configuration* node.

- c. Expand the *Security* node.
 - d. Expand the *Users and Groups* node.
 - e. Select *Users*.
2. In the right pane, move the pointer over *Operations* and select *Add*.

The following image shows the Add a new user pane that opens and includes fields where you enter a user name, a password, and a description of the user. The pane includes a Help button, an OK button to instruct the system to accept inputs, and a Cancel button to escape from the pane.

Add a new user

Name:

Password:

Description:

- a. In the Name field, type a user ID.
 - b. In the Password field, type the password associated with the user ID.
 - c. In the Description field, type a description of the user (optional).
3. Click *OK*.

The following image opens and shows a new user added to the configuration. It includes a definition of a user and a user ID and description.

Operations ▶



Users

A user is an object that can be granted or denied permissions to run iWay Business Services. A user can belong to one or more groups. Policies that specify particular rights can be associated with user.

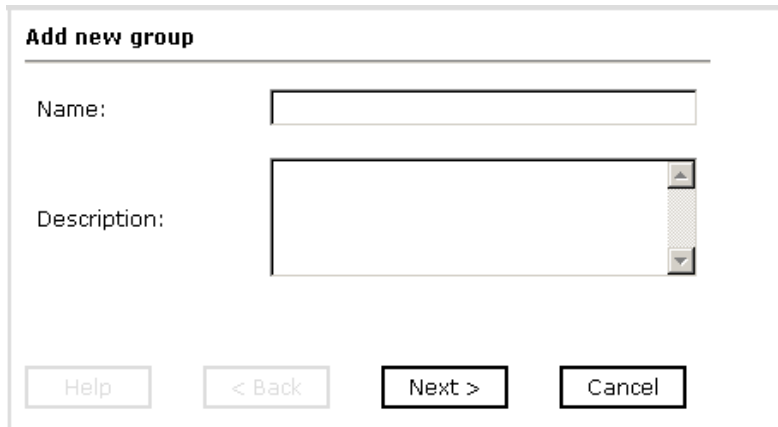
User Id	Description
<input type="checkbox"/> bse1	

Procedure: How to Create a Group to Associate With a Policy

To create a group to associate with a policy:

1. Open Servlet Application Explorer.
 - a. Click the *iWay Business Services* tab.
 - b. Expand the *Configuration* node.
 - c. Expand the *Security* node.
 - d. Expand the *Users and Groups* node.
 - e. Select *Groups*.
2. In the right pane, move the pointer over *Operations* and click *Add*.

The following image shows the Add new group pane that opens with fields where you enter a name and a description for the group. To continue after typing inputs, click the *Next* button. The pane also includes a *Help* button, a *Back* button to return to the previous screen, and a *Cancel* button to escape from the pane.



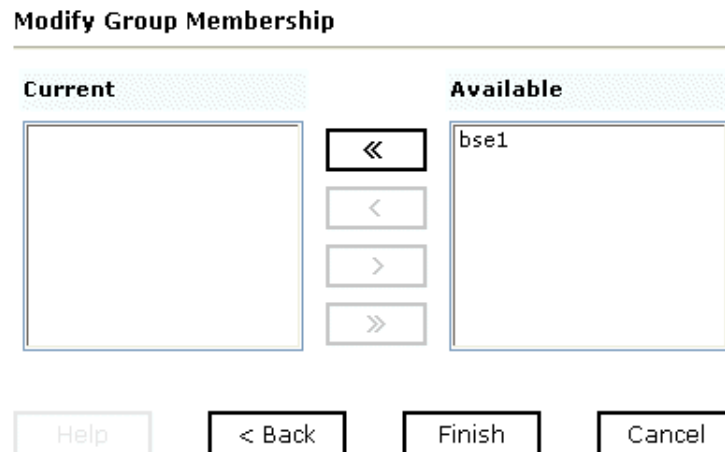
Add new group

Name:

Description:

- a. In the Name field, type a name for the group.
 - b. In the Description field, type a description for the group (optional).
3. Click *Next*.

The following image shows the Modify Group Membership pane where you can move users to or from a group using the arrow keys to move them between the Current and Available lists and then clicking the *Finish* button. The pane includes a *Help* button, a *Back* button to return to the previous screen, and a *Cancel* button to escape from the pane.



Modify Group Membership

Current **Available**

You can either highlight a single user in the list of available users and add it to the current list by clicking the left arrow, or you can click the double left arrow to add all users in the list of available users to the group.

4. After you select a minimum of one user, click *Finish*.

The new group is added.

The following image shows a pane with a new group added to the configuration. It includes a definition of a group and the group name and description.

Operations ▶



Groups

A group is an object that can be granted or denied permissions to run iWay Business Services. A group is used as a container for one or more users. Policies that specify particular rights can be associated with a group.

Group name	Description
<input type="checkbox"/> newgroup	

Procedure: How to Create an Execution Policy

To create an execution policy:

1. Open Servlet Application Explorer.
 - a. Click the *iWay Business Services* tab.
 - b. Expand the *Configuration* node.
 - c. Select *Policies*.

The following image shows the Policies pane on the right where you apply a policy. The Operations menu becomes available with three options, Build/Rebuild, Add, and Refresh.



2. Move the pointer over *Operations* and click *Add*.

The following image shows the Add a new policy pane that opens with fields for entering the name, type, and description of the policy. To continue, click the *Next* button. The pane includes a *Help* button, a *Back* button to return to the previous screen, and a *Cancel* button to escape from the pane.

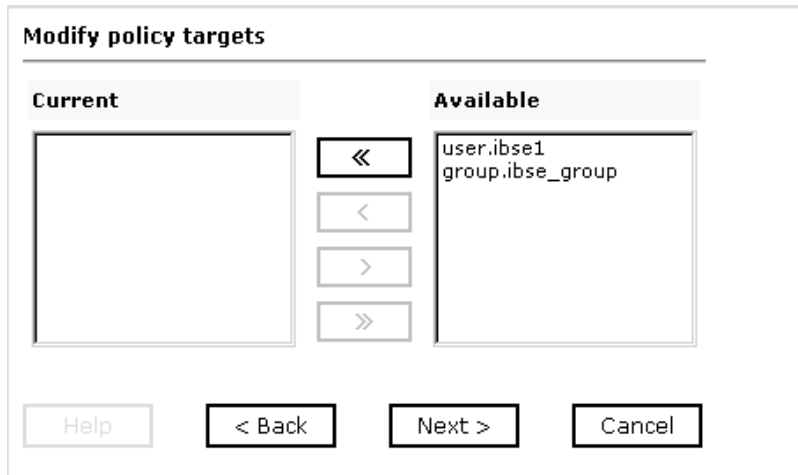
The dialog box is titled 'Add a new policy' and contains the following fields and buttons:

- Name:
- Type:
- Description:
- Buttons:

- a. In the Name field, type a a name for the policy.
- b. From the Type drop-down list, select *Execution*.
- c. In the Description field, type a description for the policy (optional).

3. Click *Next*.

The following image shows the Modify policy targets pane that opens and includes a list of current and available targets and arrow buttons to move targets from one list to the other. The pane also includes a Help button, a Back button to return to the previous screen, a Next button to continue to the next screen, and a Cancel button to escape from the pane.



4. Select a minimum of one user or group from the Available pane.

Note: This user ID is checked against the value in the user ID element of the SOAP header sent to iBSE in a SOAP request.

5. Click *Next*.

The following image shows the Modify policy permissions pane that opens and includes drop-down lists where you can select to grant or deny permission to members and then click a button to finish. The pane also includes a Help button, a Back button to return to the previous screen, and a Cancel button to escape from the pane.

Member Id	Permission
user.ibse1	Deny
group.ibse_group	Deny

Buttons: Help, < Back, Finish, Cancel

6. To assign whether users or groups may execute the iBSE, select *Grant* to permit execution or *Deny* to restrict execution from a Permission drop-down list.
7. Click *Finish*.

The following image shows the pane that summarizes your configuration. It includes a definition of policies and the name, type, and description of the policies.

Operations ▶

Policies

You can configure policies for the iWay Business Services Engine to manage resource execution, service routing, data restrictions and failover/recovery actions.

Name	Type	Description
<input type="checkbox"/> ibse_policy	Execution	

Procedure: How to Configure IP and Domain Restrictions

To configure IP and domain restrictions:

1. Open Servlet Application Explorer.

- a. Select the *iWay Business Services* tab.
 - b. Expand the *Configuration* node.
 - c. Expand the *Security* node.
 - d. Select *IP and Domain*.
2. In the right pane, move the pointer over *Operations* and click *Add*.

The following image shows the Add a new IP/Domain pane that opens where you enter information for the IP/Domain in four fields. You must select a type of restriction from a drop-down list before you can enter information in the IP(Mask)/Domain field. The pane also includes a Help button, an OK button to instruct the system to accept inputs, and a Cancel button to escape from the pane.

Add a new IP/Domain

IP(Mask)/Domain:

Type:

Access Control:

Description:

- a. From the Type drop-down list, select the type of restriction.
- b. In the IP(Mask)/Domain field, type the IP or domain name using the following guidelines.

If you select Single (Computer) from the Type drop-down list, you must provide the IP address for that computer. If you only know the DNS name for the computer, click *DNS Lookup* to obtain the IP Address based on the DNS name.

If you select Group (of Computers), you must provide the IP address and subnet mask for the computer group.

If you select Domain, you must provide the domain name, for example, yahoo.com.

3. From the Access Control drop-down list, select *Grant* to permit access or *Deny* to restrict access for the IP addresses and domain names you are adding.
4. Click *OK*.

The following image shows the pane that opens and summarizes your configuration including the domain name, whether access is granted or denied, and a description (optional).

Operations ▶



IP and Domain

You can configure the iWay Business Services Engine to use policies that control access from a single IP address, a group of IP addresses, or all addresses within a particular domain.

IP(Mask) / Domain	Access	Description
<input type="checkbox"/> test	Deny	

CHAPTER 5

Management and Monitoring

Topics:

- Managing and Monitoring Services and Events Using iBSE
- Managing and Monitoring Services and Events Using the JCA Test Tool
- Setting Engine Log Levels
- Configuring Connection Pool Sizes
- Migrating Repositories
- Exporting or Importing Targets
- Retrieving or Updating Web Service Method Connection Information
- Starting or Stopping a Channel Programmatically

After you create services and events using Servlet Application Explorer, you can use managing and monitoring tools provided by the iWay Business Services Engine (iBSE) and the iWay Connector for JCA to measure the performance of your run-time environment. This section describes how to configure and use these features.

Managing and Monitoring Services and Events Using iBSE

iWay Business Services Engine (iBSE) provides a console to manage and monitor services and events currently in use and to display resource usage and invocation statistics. These indicators can help you adjust your environment for optimum efficiency.

The following monitoring levels are available for services:

- System
- Service
- Method

The following monitoring levels are available for events:

- System
- Channel
- Port

Procedure: How to Configure Monitoring Settings

To configure monitoring settings:

1. Ensure that your application server is started.
2. To access the monitoring console, enter the following URL in your Web browser:

`http://localhost:port/ibse/IBSEConfig`

where:

`localhost`

Is the machine where the application server is running.

`port`

Is the HTTP port for the application server.

The following image shows the iBSE Settings window that opens. It lists property names and includes fields where you can enter values for each property. To configure system settings, the System pane contains drop-down lists for selecting language, encoding, the debug level, and the number of asynchronous processors. It also contains a field where you can enter a path to the adapters lib directory.

To configure security settings, the Security pane contains fields for typing the Admin User name and the associated password and a check box for specifying policy.

To configure repository settings, the Repository pane contains a drop-down list for selecting the repository type, fields to type information for the repository URL, driver, user, and password, and a check box where you can enable repository pooling. In the upper and lower right of the window is a Save button. In the lower left of the window is an option to click to access more configuration settings.

iBSE Settings:		Save
Property Name	Property Value	
System		
Language	English ▾	
Adapter Lib Directory	C:\Program Files\iWay55\lib	
Encoding	UTF-8 ▾	
Debug Level	NONE ▾	
Number of Async. Processors	0 ▾	
Security		
Admin User	iway	
Admin Password	****	
Policy	<input type="checkbox"/>	
Repository		
Repository Type	File System ▾	
Repository Url	file://C:\Program Files\iWay55\bea\ibe	
Repository Driver		
Repository User		
Repository Password		
Repository Pooling	<input type="checkbox"/>	
More configuration...		Save

3. Click *More configuration*.

Tip: To access the monitoring console directly, enter the following URL in your Web browser:

<http://localhost:port/ibse/IBSEStatus>

where:

[localhost](#)

Is the machine where the application server is running.

[port](#)

Is the HTTP port for the application server.

The following image shows the iBSE Monitoring Settings window that opens. It lists property names and includes a corresponding field where you can enter values for each property. The Monitoring pane contains a drop-down list for selecting the repository type, fields to type information for the repository URL, driver, user, and password, and a check box where you can enable repository pooling. The Auditing pane contains an option button to click to specify whether to store a message and a drop-down list where you can select the maximum messages to store. At the bottom of the window is a row of buttons that you can click to save your configuration, view events, or view services. The Save History button is inactive. After you enter properties and choose whether to save or view, you can click the Start Monitoring button.

iBSE Monitoring Settings:	
Property Name	Property Value
Monitoring	
Repository Type	File System
Repository Url	file://C:\Program Files\iWay55\bes
Repository Driver	
Repository User	
Repository Password	
Repository Pooling	<input type="checkbox"/>
Auditing	
Store Message	<input type="radio"/> yes <input checked="" type="radio"/> no
Max Message Stored	10,000
<input type="button" value="Save Configuration"/> <input type="button" value="Save History"/> <input type="button" value="View Events"/> <input type="button" value="View Services"/>	
<input type="button" value="Start Monitoring"/>	

- a. In the Monitoring pane, from the Repository Type drop-down list, select the type of repository you are using.
- b. To connect to the database in the Repository Url field, type a JDBC URL.
- c. To connect to the database in the Repository Driver field, type a JDBC Class.
- d. To access the monitoring repository database, type a user ID and password.
- e. To enable pooling, click the *Repository Pooling* check box.
- f. In the Auditing pane, select *yes* if you want to store messages.
This option is disabled by default.
Note: You must start and then, stop monitoring to enable this option.
- g. Select the maximum number of messages you want to store.

By default, 10,000 is selected.

Note: Depending on your environment and the number of messages that are exchanged, storing a large number of messages may affect system performance. If you need more information about your system resources, consult your system administrator.

- h.** Click *Save Configuration*.
- 4.** Click *Start Monitoring*.
iBSE begins to monitor all services and events currently in use. If you selected the option to store messages, iBSE stores messages.
- 5.** To stop monitoring, click *Stop Monitoring*.

Procedure: How to Monitor Services

To monitor services:

- 1.** Ensure that your application server is started.
- 2.** From the iBSE Monitoring Settings window, click *Start Monitoring*.
- 3.** Click *View Services*.

The following image shows the System Level Summary (Service Statistics) window that opens. The Web Service Methods pane contains a drop-down list where you select a service. On the right, space is reserved for a drop-down list of methods that will appear. The Statistics pane contains a table with a summary of service statistics and two drop-down lists where you can select a successful or failed invocation to view more information about that service. At the bottom of the window is a home button to click to return to the iBSE Monitoring Settings window.

The screenshot shows a window titled "Service Statistics". It is divided into two main sections: "Web Service Methods" and "Statistics".

Web Service Methods: This section contains a "Service" label and a "Method" label. Below the "Service" label is a drop-down menu currently showing "all".

Statistics: This section contains a table with the following data:

Total Time	55 min
Total Request Count	1
Total Success Count	1
Total Error Count	0
Average Request Size	409.0 bytes
Average Response Size	665.0 bytes
Average Execution Time	656 ms
Last Execution Time	828 ms
Average Back End Time	530 ms
Last Back End Time	765 ms
Successful Invocations	select a correlation id
Failed Invocations	select a correlation id

At the bottom right of the window is a button labeled "< home".

The system level summary provides services statistics at a system level.

The following table consists of two columns, one that lists the name of each statistic and the other that describes the corresponding service statistic.

Statistic	Description
Total Time	Total amount of time iBSE monitors services. The time starts after you click Start Monitoring in the iBSE Monitoring Settings window.

Statistic	Description
Total Request Count	Total number of services requests that were made during the monitoring session.
Total Success Count	Total number of successful service executions.
Total Error Count	Total number of errors that were encountered.
Average Request Size	Average size of an available service request.
Average Response Size	Average size of an available service response size.
Average Execution Time	Average execution time for a service.
Last Execution Time	Last execution time for a service.
Average Back End Time	Average back end time for a service.
Last Back End Time	Last back end time for a service.
Successful Invocations	A list of successful services arranged by correlation ID. To retrieve more information for a service, you can select the service from the drop-down list.
Failed Invocations	A list of failed services arranged by correlation ID. To retrieve more information for a service, you can select the service from the drop-down list.

4. Select a service from the drop-down list.

The following image shows the System Level Summary (Service Statistics) window that opens. The Web Service Methods pane contains a drop-down list on the left where you select a service and a drop-down list on the right where you select a service method. The Statistics pane contains a table with a summary of service statistics and two drop-down lists. To view more information about that service, you can select it from the Successful Invocations or Failed Invocations drop-down list. To suspend or resume a service, you can click a button in the lower right. To return to the iBSE Monitoring Settings window, you click the home button (also located in the lower right).

Service Statistics

Web Service Methods

Service

Method

BD100033 ▾

all methods ▾

Statistics

Total Time	1 hrs
Total Request Count	1
Total Success Count	1
Total Error Count	0
Average Request Size	409.0 bytes
Average Response Size	665.0 bytes
Average Execution Time	656 ms
Last Execution Time	656 ms
Average Back End Time	530 ms
Last Back End Time	530 ms
Successful Invocations	select a correlation id ▾
Failed Invocations	select a correlation id ▾

Suspend Service

< home

- a. To stop a service at any time, click *Suspend Service*.
- b. To restart the service, click *Resume Service*.
5. Select a method for the service from the Method drop-down list.

The following image shows the Method Level Summary (Service Statistics) window that opens. The Web Service Methods pane contains a drop-down list on the left where you select a service and a drop-down list on the right where you select a service method. The Statistics pane contains a table with a summary of service statistics and two drop-down lists. To view more information about that service, you can select it from the Successful Invocations or Failed Invocations drop-down list. To suspend or resume a service, you can click a button in the lower right. To return to the iBSE Monitoring Settings window, you click the home button (also located in the lower right).

The screenshot shows a window titled "Service Statistics". It is divided into two main sections: "Web Service Methods" and "Statistics".

Web Service Methods

Service: [B0100033 ▼]
Method: [GetEffectiveAddress ▼]

Statistics

Total Time	1 hrs
Total Request Count	1
Total Success Count	1
Total Error Count	0
Average Request Size	409.0 bytes
Average Response Size	665.0 bytes
Average Execution Time	656 ms
Last Execution Time	656 ms
Average Back End Time	530 ms
Last Back End Time	530 ms
Successful Invocations	[select a correlation id ▼]
Failed Invocations	[select a correlation id ▼]

[Suspend Service]

[< home]

6. For additional information about a successful service and its method, select a service based on its correlation ID from the Successful Invocation drop-down list.

The following image shows the Invocation Level Statistics window that opens. The Message Information pane contains a table of information about the message. The Client Information pane contains a table of information about the client. The Detail pane contains a table that shows the size of the request and response messages, with options to click to view the respective XML documents. In the lower right of the window is a home button to click to return to the iBSE Monitoring Settings window.

Invocation Statistics

Message Information

Received	2004-09-14 12:04:16.312
Sent to adapter	2004-09-14 12:04:16.406
Received from adapter	2004-09-14 12:04:16.936
Responded	2004-09-14 12:04:16.968
Status	SUCCESS

Client Information

Client IP	127.0.0.1
Client Host Name	127.0.0.1
User Name	

Detail

Message	Size
Request Message	409 bytes
Response Message	665 bytes

< home

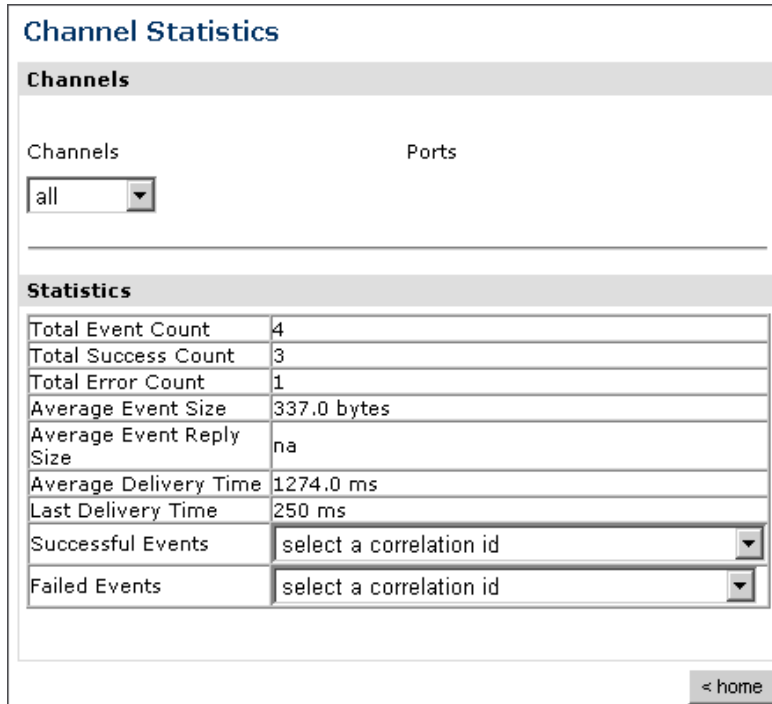
7. To view the XML request document in your Web browser, click *Request Message*.
You can also view the XML response document for the service.
8. To return to the iBSE Monitoring Settings window, click *home*.

Procedure: How to Monitor Events

To monitor events:

1. Ensure that your application server is started.
2. In the iBSE Monitoring Settings window, click *Start Monitoring*.
3. Click *View Events*.

The following image shows the System Level Summary (Channel Statistics) window that opens. The Channels pane contains a drop-down list on the left where you select a channel. On the right, space is reserved for a drop-down list of ports that will appear. The Statistics pane contains a table with a summary of event statistics and two drop-down lists where you can select a successful or failed event to view more information about that event. In the lower right of the window is a home button to click to return to the iBSE Monitoring Settings window.



The system level summary provides event statistics at a system level.

The following table consists of two columns, one that lists the name of each statistic and the other that describes the corresponding event statistic.

Statistic	Description
Total Event Count	Total number of events.
Total Success Count	Total number of successful event executions.
Total Error Count	Total number of errors that were encountered.
Average Event Size	Average size of an available event request.

Statistic	Description
Average Event Reply Size	Average size of an available event response.
Average Delivery Time	Average delivery time for an event.
Last Delivery Time	Last delivery time for an event.
Successful Events	List of successful events arranged by correlation ID. To retrieve more information for an event, select the event from the drop-down list.
Failed Events	List of failed events arranged by correlation ID. To retrieve more information for an event, select the event from the drop-down list.

4. Select a channel from the drop-down list.

The following image shows the Channel Level Event Summary (Channel Statistics) window that opens. The Channels pane contains a drop-down list on the left where you select a channel and a drop-down list on the right where you select a port. The Statistics pane contains a table with a summary of event statistics and two drop-down lists where you can select a successful or failed event to view more information about that event. In the lower right of the window is a button to click to suspend or resume a channel and a home button to click to return to the iBSE Monitoring Settings window.

The screenshot shows a window titled "Channel Statistics". It has two main sections: "Channels" and "Statistics".

Channels Section:

- Channels: TestChan (dropdown)
- Ports: all (dropdown)

Statistics Section:

Total Event Count	3
Total Success Count	2
Total Error Count	1
Average Event Size	401.0 bytes
Average Event Reply Size	na
Average Delivery Time	1542.0 ms
Last Delivery Time	250 ms
Successful Events	select a correlation id (dropdown)
Failed Events	select a correlation id (dropdown)

Control Buttons:

- Suspend Channel
- Start Channel
- < home

- a. To stop a channel at any time, click *Suspend Channel*.
- b. To start the channel, click *Start Channel*.
5. From the Ports drop-down list, select a port for the channel.

The following image shows the Port Level Event Summary (Channel Statistics) window that opens. The Channels pane contains a drop-down list on the left where you select a channel and a drop-down list on the right where you select a port. The Statistics pane contains a table with a summary of event statistics and two drop-down lists where you can select a successful or failed event to view more information about that event. In the lower right of the window is a button to click to suspend or resume a channel and a home button to click to return to the iBSE Monitoring Settings window.

Channel Statistics

Channels

Channels: TestChan ▾ Ports: TestPort ▾

Statistics

Total Event Count	2
Total Success Count	2
Total Error Count	0
Average Event Size	446.0 bytes
Average Event Reply Size	na
Average Delivery Time	2189.0 ms
Last Delivery Time	na
Successful Events	select a correlation id ▾
Failed Events	select a correlation id ▾

- For more information about a successful event and its port, select an event based on its correlation ID from the Successful Events drop-down list.

The following image shows the Event Level Statistics (Message Statistics) window that opens. The Message Information pane contains a table of information pertaining to the event message. The Messages pane contains a table that shows the size of the event and reply messages, with an option to view an XML document of the event message. In the lower right of the window is a home button to click to return to the iBSE Monitoring Settings window.

The screenshot displays a window titled "Message Statistics". It is divided into two main sections: "Message Information" and "Messages".

Message Information

Received At	2004-09-14 12:18:20.842
Disposed At	● TestPort
Delivered At	2004-09-14 12:18:23.562

Messages

Detail	size
Event Message	446 bytes
Reply Message	na

In the bottom right corner of the window, there is a button labeled "< home".

- a. To view the XML event document in your Web browser, click *Event Message*.
- b. To return to the iBSE Monitoring Settings window, click *home*.

Managing and Monitoring Services and Events Using the JCA Test Tool

The JCA Test Tool, which is also known as the JCA Installation Verification Program (IVP), provides a console to manage and monitor services and events currently in use and to display resource usage and invocation statistics. These indicators can help you adjust your environment for optimum efficiency.

Procedure: How to Manage and Monitor Services Using the JCA Test Tool

To manage and monitor services using the JCA Test Tool:

1. Open a Web browser to:

<http://localhost:port/iwjcaivp>

where:

[localhost](#)

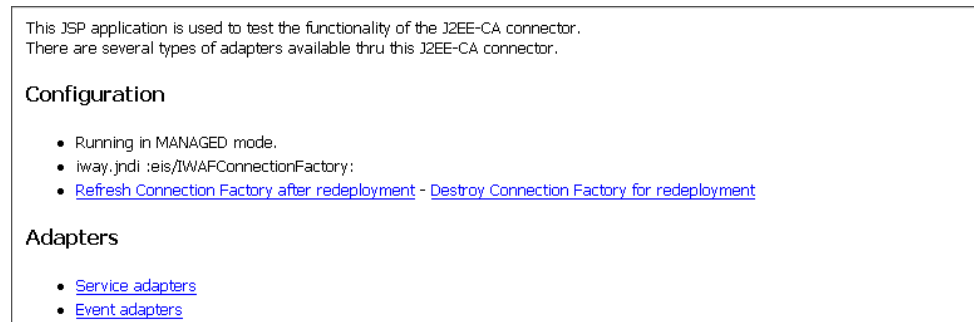
Is the name of the machine where your application server is running.

[port](#)

Is the HTTP port for the application server, for example:

<http://localhost:7001/iwjcaivp>

The following image shows the JCA Test Tool page that opens. The page contains a description of the function of the tool and configuration information, including options to change your connection settings. It also provides options for viewing service or event adapters.



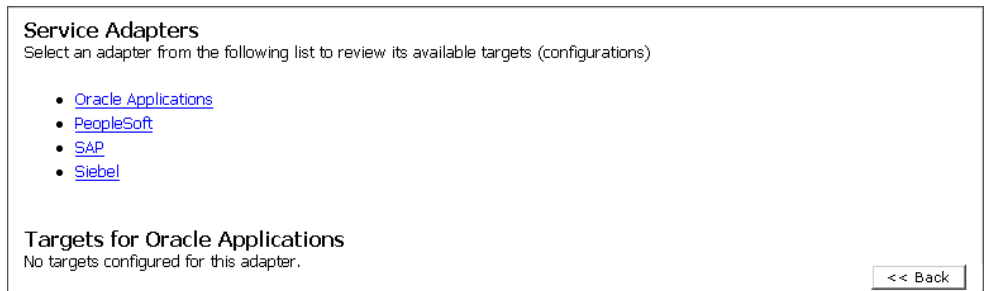
The JCA Test Tool runs in managed mode by default.

2. Perform the following steps to monitor the latest service adapter configuration.

Note: You must perform these steps for every new adapter target that is created using a JCA implementation of Application Explorer. In addition, you also must perform these steps for every new JCA configuration that is created using Application Explorer.

- a. Click *Destroy Connection Factory for redeployment*.
 - b. Redeploy the JCA connector.
 - c. In the JCA Test Tool, click *Refresh Connection Factory after redeployment*.
3. Click *Service adapters*.

The following image shows the Service Adapters page that opens. The page provides a live list of available service adapters and a list of targets configured for a specific adapter. In the lower right is a Back button to click to return to the previous page.



4. Select a service adapter to monitor.

The following image shows the page that opens. The left side provides a live list of available service adapters and a list of any targets configured for a specific adapter. The upper right side shows statistics for a selected target. The middle right has a User field and a Password field. The lower right contains a box where you type or paste an input document. Below the input box is a Send button to click to send a request for a test service and a Reset button to click to reset the fields. In the lower right is a Back button to click to return to the previous page.

<p>Service Adapters Select an adapter from the following list to review its available targets (configurations)</p> <ul style="list-style-type: none"> • Oracle Applications • PeopleSoft • SAP • Siebel 	<p>Statistics for Siebel target TestService</p> <p>TotalRequestCount : 0 TotalSuccessCount : 0 TotalErrorCount : 0 AverageExecutionTime : 0 msec. LastExecutionTime : 0 msec.</p>
<p>Targets for Siebel</p> <ul style="list-style-type: none"> • TestService 	<p>Request for Siebel target TestService Enter the data for this interaction. The configured user/password will be used if the User name is not provided.</p> <p>User: <input type="text"/></p> <p>Password: <input type="password"/></p> <p>Input Doc: <input type="text"/></p> <p><input type="button" value="Send"/> <input type="button" value="Reset"/></p> <p style="text-align: right;"><input type="button" value=" << Back"/></p>

- a. Click the desired target for your service adapter.
 - b. In the Request area, enter a user name and password.
 - c. In the Input Doc area, enter a request document that was created from the request schema for your service.
5. Click *Send*.

The following image shows the updated statistics that appear for your service if the request is successful. The statistics include the total number of requests, successes, and errors and the average and last execution time in milliseconds.

TotalRequestCount	: 0
TotalSuccessCount	: 0
TotalErrorCount	: 0
AverageExcectionTime	: 0 msec.
LastExcectionTime	: 0 msec.

Procedure: How to Manage and Monitor Events Using the JCA Test Tool

To manage and monitor events using the JCA Test Tool:

1. Open a Web browser to:

<http://localhost:port/iwjcaivp>

where:

localhost

Is the name of the machine where your application server is running.

port

Is the HTTP port for the application server, for example:

<http://localhost:7001/iwjcaivp>

The following image shows the JCA Test Tool page that opens. The page contains a description of the function of the tool and configuration information, including options to change your connection settings. It also provides options for viewing service or event adapters.

This JSP application is used to test the functionality of the J2EE-CA connector. There are several types of adapters available thru this J2EE-CA connector.

Configuration

- Running in MANAGED mode.
- `iwjca.jndi :eis/IWAFConnectionFactory`:
- [Refresh Connection Factory after redeployment](#) - [Destroy Connection Factory for redeployment](#)

Adapters

- [Service adapters](#)
- [Event adapters](#)

The JCA Test Tool runs in managed mode by default.

2. Perform the following steps to monitor the latest event adapter configuration.

Note: You must perform these steps for every new adapter target that is created using a JCA implementation of Application Explorer. In addition, you must also perform these steps for every new JCA configuration that is created using Application Explorer.

- a. Click *Destroy Connection Factory for redeployment*.
 - b. Redeploy the JCA connector.
 - c. In the JCA Test Tool, click *Refresh Connection Factory after redeployment*.
3. Click *Event adapters*.
The Event Adapters page opens.
 4. Select the event adapter to monitor.
 5. Click the desired channel for your event adapter.
 6. Click *start*.

The following image shows the updated statistics for your channel and the port. The statistics include the total number of requests, successes, and errors and the average and last execution time in milliseconds. There are options to click in the upper right of the page to start or refresh the channel.

Current channel Statistics	
Commands: start refresh	
Active: false	
TotalRequestCount	: 0
TotalSuccessCount	: 0
TotalErrorCount	: 0
AverageExcecutionTime	: 0 msec.
LastExcecutionTime	: 0 msec.
Statistics for port 'fileIN'	
TotalRequestCount	: 0
TotalSuccessCount	: 0
TotalErrorCount	: 0
AverageExcecutionTime	: 0 msec.
LastExcecutionTime	: 0 msec.

Setting Engine Log Levels

The following section describes how to set engine log levels for Servlet iBSE and JCA. For more information, see the documentation.

Procedure: How to Enable Tracing for Servlet iBSE

To enable tracing for Servlet iBSE:

1. Open the Servlet iBSE configuration page at:

`http://localhost:port/ibse/IBSEConfig`

where:

`localhost`

Is the name of the machine where your application server is running.

`port`

Is the HTTP port for the application server, for example:

`http://localhost:7001/ibse/IBSEConfig`

2. In the System pane, from the Debug drop-down list, select the level of tracing.
3. Click *Save*.

Tracing information is written to the `ibselogs` directory where your application server accesses or has expanded Servlet iBSE.

Procedure: How to Enable Tracing for JCA

To enable tracing for JCA:

1. Open the extracted `ra.xml` file in a text editor.
2. Locate and change the following setting:

LogLevel. This setting can be set to `DEBUG`, `INFO`, or `ERROR`.

```
<context-param>
<config-property>
  <config-property-name>LogLevel</config-property-name>
  <config-property-type>java.lang.String</config-property-type>
  <config-property-value></config-property-value>
</config-property>
```

For example:

```
<config-property-value>DEBUG</config-property-value>
```

A directory in the configuration directory contains the logs.

- a. Review the logs generated by your application server.
 - b. Leave the remainder of the previous file unchanged.
3. Save the file and exit the editor.
 4. Redeploy the connector.

Configuring Connection Pool Sizes

The following topic describes how to configure connection pool sizes for the JCA connector.

Procedure: How to Configure Connection Pool Sizes

To configure connection pool sizes:

1. Open the extracted ra.xml file in a text editor.
2. Locate and change the following setting:

pool-params. The JCA Resource Connector has an initial capacity value of 0 by default and cannot be changed. The maximum capacity value is 10 by default and can be changed to a higher value.

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE weblogic-connection-factory-dd (View Source for full
doctype...)>
- <weblogic-connection-factory-dd>
  <connection-factory-name>IWAFJCA</connection-factory-name>
  <jndi-name>eis/IWAFConnectionFactory</jndi-name>
  - <pool-params>
    <initial-capacity>0</initial-capacity>
    <max-capacity>10</max-capacity>
    <capacity-increment>1</capacity-increment>
    <shrinking-enabled>>false</shrinking-enabled>
    <shrink-period-minutes>200</shrink-period-minutes>
  </pool-params>
  <security-principal-map />
</weblogic-connection-factory-dd>
```

3. Save the file and exit the editor.
4. Redeploy the connector.

Migrating Repositories

During design time, a repository is used to store metadata created when using Application Explorer to configure adapter connections, browse EIS objects, configure services, and configure listeners to listen for EIS events. For more information on configuring repositories, see the *iWay 5.5 Installation and Configuration* documentation.

The information in the repository also is referenced at run time. For management purposes, you can migrate iBSE and JCA repositories to new destinations without affecting your existing configuration. For example, you may want to migrate a repository from a development environment to a production environment.

File Repositories

If you want to migrate a File repository to another destination, copy the `ibserrepo.xml` file from the following path:

```
drive:\Program Files\iWay55\ibse\ibserrepo.xml
```

where:

```
drive
```

Is the location of your iWay 5.5 installation.

You can place the `ibserrepo.xml` file in a new location that is a root directory of the iBSE Web application, for example:

```
drive:\ProductionConfig\ibse\ibserrepo.xml
```

iBSE Repositories

The following topic describes how to migrate an iBSE repository that is configured for Oracle. You can follow the same procedure if you want to migrate an iBSE repository that is configured for Microsoft SQL Server 2000, Sybase, or DB2. However, when you are configuring a new environment, you must execute the script that creates the repository tables for your database. In addition, verify that all required files and drivers for your database are in the class path. For more information on configuring repositories, see the *iWay 5.5 Installation and Configuration* documentation.

Note: The following procedure allows you to migrate only Web services. If migrating event handling information is one of your requirements, you must migrate at the database level. For more information, see *Migrating Event Handling Configurations* on page 5-28.

Procedure: How to Migrate an iBSE Repository Configured for Oracle

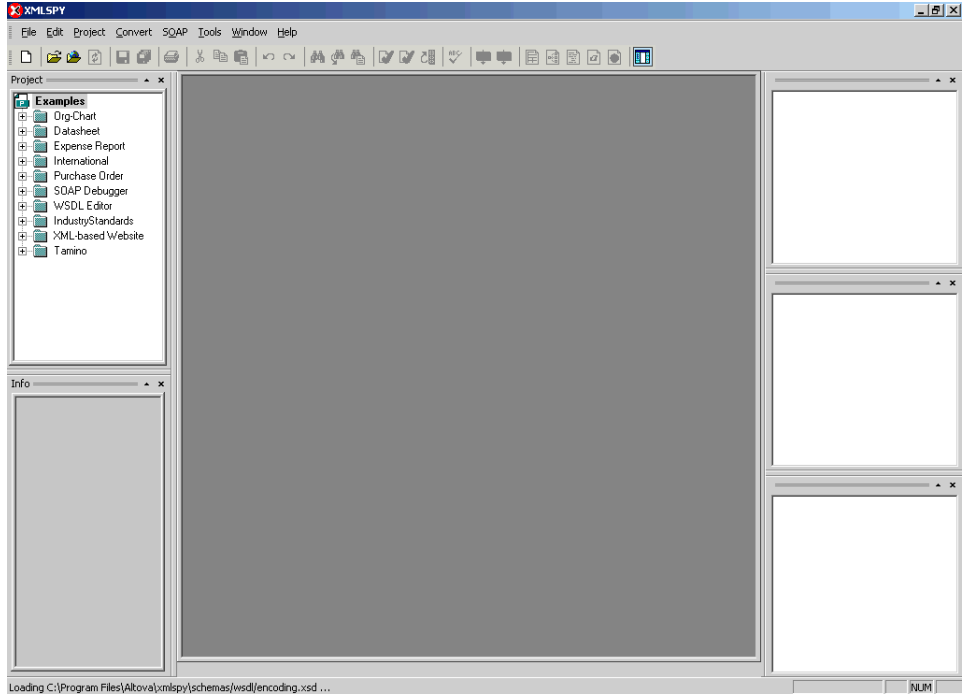
To migrate an iBSE repository that is configured for Oracle:

1. Copy the iBSE configuration service URL, for example:

```
http://localhost:7777/ibse/IBSEServlet/admin/iwconfig.ibs?wsdl
```

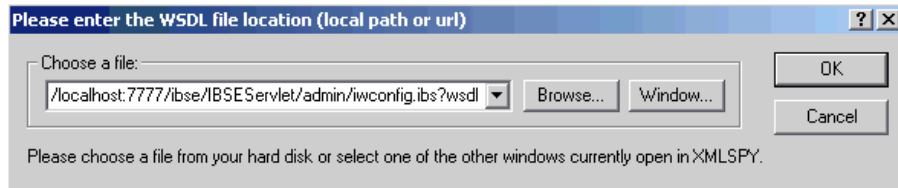
2. Open a third party XML editor, for example, XMLSPY.

The following image shows the XMLSPY window. The upper left has a Project pane that contains a tree of sample files, and the lower left has a blank Info pane. The middle pane is blank. The right side is divided into three blank panes.



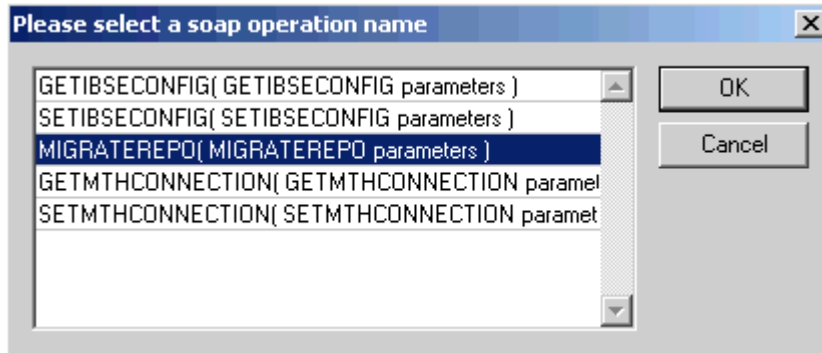
3. From the SOAP menu, select *Create new SOAP request*.

The following image shows the WSDL file location dialog box that opens, where you enter a local path or URL. The dialog includes Browse, Window, OK, and Cancel buttons.



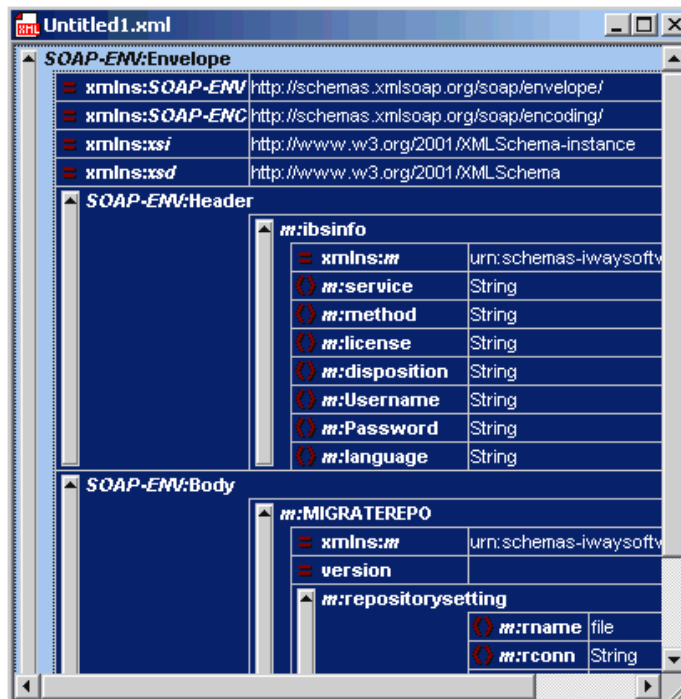
4. In the Choose a file field, paste the iBSE configuration service URL.
5. Click *OK*.

The following image shows the soap operation name dialog box that opens with a list of available control methods. You can select from the list and click *OK* or to escape from the dialog box, you can click *Cancel*.



6. Select the *MIGRATEREPO(MIGRATEREPO parameters)* control method and click *OK*.

The following image shows a portion of the window that opens with the structure of the SOAP envelope. It includes information about location and schemas.



7. Locate the *Text* view icon in the tool bar.

In the following image, the pointer points to the Text view icon.



8. To display the structure of the SOAP envelope as text, click the *Text view* icon.

The <SOAP-ENV:Header> tag is not required and can be deleted from the SOAP envelope.

9. Locate the following section:

```
<m:MIGRATEREPO
xmlns:m="urn:schemas-iwaysoftware-com:jul2003:ibse:config" version="">
<m:repositorysetting>
<m:rname>oracle</m:rname>
<m:rconn>String</m:rconn>
<m:rdriver>String</m:rdriver>
<m:ruser>String</m:ruser>
<m:rpwd>String</m:rpwd>
</m:repositorysetting>
<m:servicename>String</m:servicename>
</m:MIGRATEREPO>
```

- a. For the <m:rconn> tag, replace the String placeholder with the repository URL where you want to migrate your existing iBSE repository.

For example, the Oracle repository URL has the following format:

```
jdbc:oracle:thin:@[host]:[port]:[sid]
```

- b. For the <m:rdriver> tag, replace the String placeholder with the location of your Oracle driver.

Note: This is an optional tag. If you do not specify a value, the default Oracle JDBC driver is used.

- c. For the <m:ruser> tag, replace the String placeholder with a valid user name to access the Oracle repository.
- d. For the <m:rpwd> tag, replace the String placeholder with a valid password to access the Oracle repository.

10. Perform one of the following migration options.

If you want to migrate a **single** Web service from the current iBSE repository, enter the Web service name in the <m:servicename> tag, for example:

```
<m:servicename>Service1</m:servicename>
```

If you want to migrate **multiple** Web services from the current iBSE repository, duplicate the `<m:servicename>` tag for each Web service, for example:

```
<m:servicename>Service1</m:servicename>  
<m:servicename>Service2</m:servicename>
```

If you want to migrate **all** Web services from the current iBSE repository, remove the `<m:servicename>` tag.

11. From the SOAP menu, select *Send request to server*.

Your iBSE repository and the Web services you specified migrate to the new Oracle repository URL that you specified.

JCA Repositories

The following procedure describes how to migrate a JCA repository. For more information on configuring JCA repositories, see the documentation.

Procedure: How to Migrate a JCA Repository

To migrate a JCA repository:

1. Navigate to the location of your JCA configuration directory where the repository schemas and other information is stored, for example:

```
C:\Program Files\iWay55\config\base
```
2. Locate and copy the *repository.xml* file.
3. Place this file in a new JCA configuration directory to migrate the existing repository.

Your JCA repository migrates to the new JCA configuration directory.

Migrating Event Handling Configurations

This topic describes how to migrate your iBSE repositories at a database level for Microsoft SQL Server 2000, Oracle, Sybase, or DB2. You can use this information to migrate event handling information, for example, port or channel configurations.

Procedure: How to Migrate a Microsoft SQL Server 2000 Repository

To migrate a Microsoft SQL Server 2000 repository:

1. Open a command prompt and navigate to the iWay setup directory. The default location on Windows is:

```
C:\Program Files\iWay55\etc\setup
```

This directory contains SQL to create the repository tables in the following file:

```
iwse.sql
```


You can use `iwse.sql` to create the database tables that are used by iBSE. For example, the following image shows the tree in the left pane and tables in the right pane. The tables are listed by name in one column with corresponding columns for information about owner, type, and the date the table was created.

The screenshot shows the Microsoft SQL Server Enterprise Manager interface. The left pane displays a tree view of the server structure, with 'ibserrepos' selected under 'Databases'. The right pane shows a list of 47 tables with columns for Name, Owner, Type, and Create Date.

Name	Owner	Type	Create Date
AF_Bindings	dbo	User	8/4/2004 12:56:00 PM
AF_Channels	dbo	User	8/4/2004 12:56:00 PM
AF_CONFIG	dbo	User	8/4/2004 12:56:05 PM
AF_Keys	dbo	User	8/4/2004 12:56:00 PM
AF_Ports	dbo	User	8/4/2004 12:56:00 PM
AF_Targets	dbo	User	8/4/2004 12:56:00 PM
ChannelState	dbo	User	8/4/2004 12:56:00 PM
IBS_datasources	dbo	User	8/4/2004 12:56:02 PM
IBS_Group	dbo	User	8/4/2004 12:56:00 PM
IBS_Group_User	dbo	User	8/4/2004 12:56:00 PM
IBS_IPDomain_Type	dbo	User	8/4/2004 12:56:01 PM
IBS_IPDomainPermission	dbo	User	8/4/2004 12:56:01 PM
IBS_licenseGroup	dbo	User	8/4/2004 12:56:01 PM
IBS_licenses	dbo	User	8/4/2004 12:56:02 PM
IBS_methods	dbo	User	8/4/2004 12:56:02 PM
IBS_Object	dbo	User	8/4/2004 12:56:01 PM
IBS_Object_Policy	dbo	User	8/4/2004 12:56:01 PM
IBS_permissions	dbo	User	8/4/2004 12:56:02 PM
IBS_Policy	dbo	User	8/4/2004 12:56:01 PM
IBS_Policy_Member_Action	dbo	User	8/4/2004 12:56:01 PM
IBS_Policy_Type	dbo	User	8/4/2004 12:56:01 PM
IBS_Policy_LUGP_Type	dbo	User	8/4/2004 12:56:01 PM
IBS_services	dbo	User	8/4/2004 12:56:02 PM
IBS_servicesState	dbo	User	8/4/2004 12:56:02 PM
IBS_SOAP_servers	dbo	User	8/4/2004 12:56:01 PM
IBS_States	dbo	User	8/4/2004 12:56:02 PM
IBS_User	dbo	User	8/4/2004 12:56:02 PM
IBS_web_servers	dbo	User	8/4/2004 12:56:02 PM

For more information on configuring the Microsoft SQL Server 2000 repository, see the documentation.

2. To migrate the tables that were created by the `iwse.sql` script for iBSE, use your Microsoft SQL Server 2000 database tool set. For more information, consult your database administrator.

Procedure: How to Migrate an Oracle Repository

To migrate an Oracle repository:

1. Open a command prompt and navigate to the iWay setup directory. The default location on Windows is:

```
C:\Program Files\iWay55\etc\setup
```

This directory contains SQL to create the repository tables in the following files:

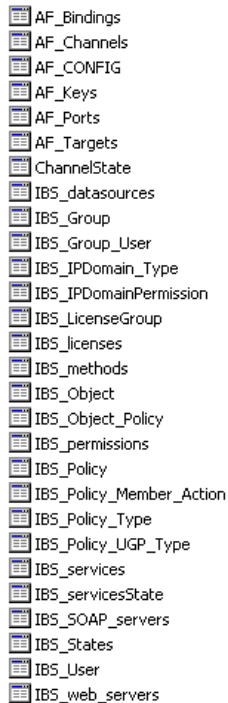
For Oracle 8:

```
iwse.ora
```

For Oracle 9:

[iwse.ora9](#)

2. To create the Oracle database tables that are used by iBSE, use the SQL script as shown in the example in the following image that shows a list of tables.



AF_Bindings
AF_Channels
AF_CONFIG
AF_Keys
AF_Ports
AF_Targets
ChannelState
IB5_datasources
IB5_Group
IB5_Group_User
IB5_IPDomain_Type
IB5_IPDomainPermission
IB5_LicenseGroup
IB5_licenses
IB5_methods
IB5_Object
IB5_Object_Policy
IB5_permissions
IB5_Policy
IB5_Policy_Member_Action
IB5_Policy_Type
IB5_Policy_UGP_Type
IB5_services
IB5_servicesState
IB5_SOAP_servers
IB5_States
IB5_User
IB5_web_servers

For more information on configuring the Oracle repository, see the [documentation](#).

3. To migrate the tables that were created by the SQL script for iBSE, use your Oracle database tool set. For more information, consult your database administrator.

Procedure: How to Migrate a Sybase Repository

To migrate a Sybase repository:

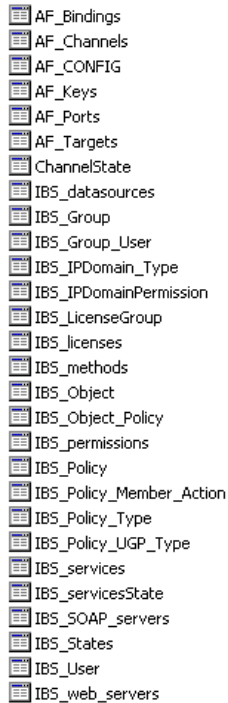
1. Open a command prompt and navigate to the iWay setup directory. The default location on Windows is:

`C:\Program Files\iWay55\etc\setup`

This directory contains SQL to create the repository tables in the following file:

`sybase-iwse.sql`

2. To create the Sybase database tables that are used by iBSE, use the SQL script as shown in the example in the following image that shows a list of tables.



A screenshot of a database table list showing various tables for AF and IB5 components. The tables listed are:

- AF_Bindings
- AF_Channels
- AF_CONFIG
- AF_Keys
- AF_Ports
- AF_Targets
- ChannelState
- IB5_datasources
- IB5_Group
- IB5_Group_User
- IB5_IPDomain_Type
- IB5_IPDomainPermission
- IB5_LicenseGroup
- IB5_licenses
- IB5_methods
- IB5_Object
- IB5_Object_Policy
- IB5_permissions
- IB5_Policy
- IB5_Policy_Member_Action
- IB5_Policy_Type
- IB5_Policy_UGP_Type
- IB5_services
- IB5_servicesState
- IB5_SOAP_servers
- IB5_States
- IB5_User
- IB5_web_servers

For more information on configuring the Sybase repository, see the [documentation](#).

3. To migrate the tables that were created by the SQL script for iBSE, use your Sybase database tool set. For more information, consult your database administrator.

Procedure: How to Migrate a DB2 Repository

To migrate a DB2 repository:

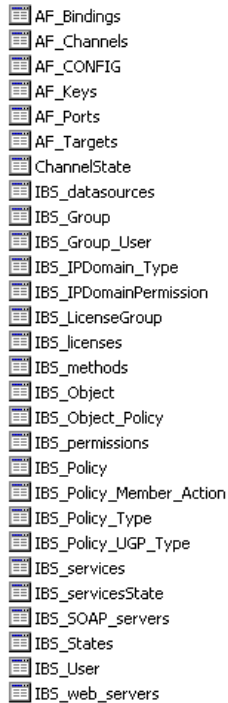
1. Open a command prompt and navigate to the iWay setup directory. The default location on Windows is:

```
C:\Program Files\iWay55\etc\setup
```

This directory contains SQL to create the repository tables in the following file:

```
db2-ibse.sql
```

2. To create the DB2 database tables that are used by iBSE, use the SQL script as shown in the example in the following image that shows a list of tables.



AF_Bindings
AF_Channels
AF_CONFIG
AF_Keys
AF_Ports
AF_Targets
ChannelState
IB5_datasources
IB5_Group
IB5_Group_User
IB5_IPDomain_Type
IB5_IPDomainPermission
IB5_LicenseGroup
IB5_licenses
IB5_methods
IB5_Object
IB5_Object_Policy
IB5_permissions
IB5_Policy
IB5_Policy_Member_Action
IB5_Policy_Type
IB5_Policy_UGP_Type
IB5_services
IB5_servicesState
IB5_SOAP_servers
IB5_States
IB5_User
IB5_web_servers

For more information on configuring the DB2 repository, see the [documentation](#).

You can migrate the tables that were created by the SQL script for iBSE using your DB2 database toolset. For more information, consult your database administrator.

Exporting or Importing Targets

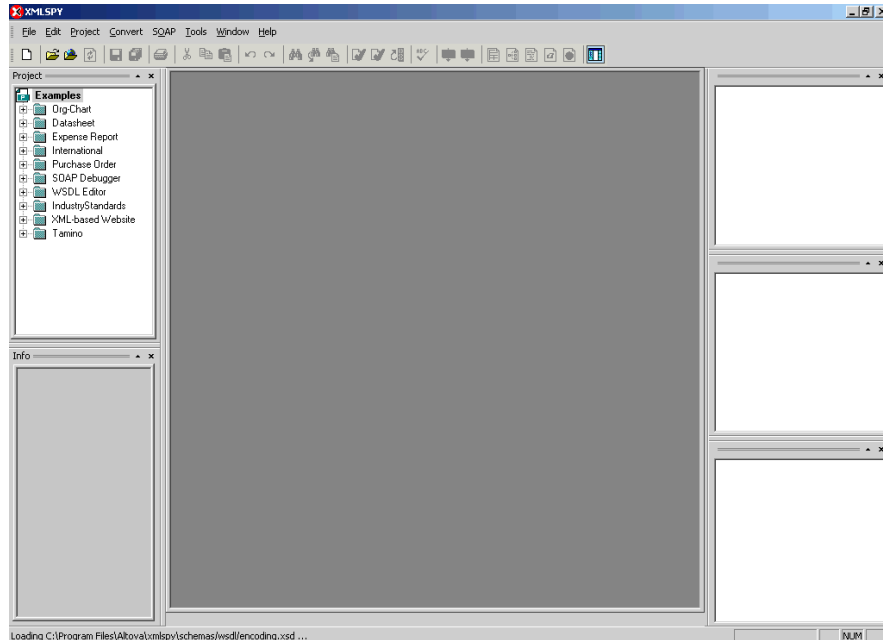
After you migrate your repository, you can export or import targets with their connection information and persistent data between repositories.

Procedure: How to Export a Target

To export a target:

1. Copy the iBSE administrative services for Application Explorer URL, for example:
<http://localhost:7777/ibse/IBSEServlet/admin/iwae.ibs?wsdl>
2. Open a third party XML editor, for example, XMLSPY.

The following image shows the XMLSPY window. The upper left has a Project pane that contains a tree of sample files, and the lower left has a blank Info pane. The middle pane is blank. The right side is divided into three blank panes.



3. From the SOAP menu, select *Create new SOAP request*.
The WSDL file location dialog box opens.
4. In the Choose a file field, paste the iBSE administrative services for Application Explorer URL.
5. Click *OK*.
The soap operation name dialog box opens and lists the available control methods.
6. Select the *EXPORTTARGET(EXPORTTARGET parameters)* control method and click *OK*.
A window opens that shows the structure of the SOAP envelope.
7. Locate the *Text view* icon in the tool bar.
In the following image, the pointer points to the Text view icon.



8. To display the structure of the SOAP envelope as text, click the *Text view* icon.

The <SOAP-ENV:Header> tag is not required and can be deleted from the SOAP envelope.

9. Locate the following section:

```
<m:EXPORTTARGET  
xmlns:m="urn:schemas-iwaysoftware-com:dec2002:iwse:af">  
<m:target>String</m:target>  
<m:name>String</m:name>  
</m:EXPORTTARGET>
```

- a. For the <m:target> tag, replace the String placeholder with the EIS target system name as it appears in Application Explorer and verify whether this value is case sensitive.
 - b. For the <m:name> tag, replace the String placeholder with the name of the target you want to export.
10. From the SOAP menu, select *Send request to server*.

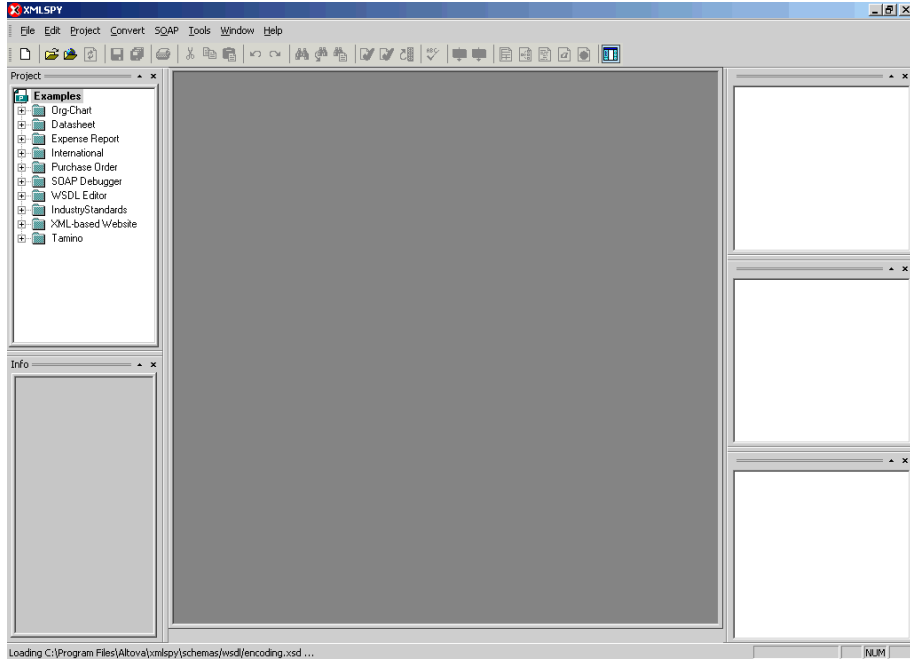
A response is returned that contains the <m: exporttime> and <m: contents> elements. You must use these elements when importing your target.

Procedure: How to Import a Target

To import a target:

1. Copy the iBSE administrative services for Application Explorer URL, for example:
<http://localhost:7777/ibse/IBSEServlet/admin/iwae.ibs?wsdl>
2. Open a third party XML editor, for example, XMLSPY.

The following image shows the XMLSPY window. The upper left has a Project pane that contains a tree of sample files, and the lower left has a blank Info pane. The middle pane is blank. The right side is divided into three blank panes.



3. From the SOAP menu, select *Create new SOAP request*.

The WSDL file location dialog box opens.

4. In the Choose a file field, paste the iBSE administrative services for Application Explorer URL and click *OK*.

The soap operation name dialog box opens and lists the available control methods.

5. Select the *IMPORTTARGET(IMPORTTARGET parameters)* control method and click *OK*.

A window opens, which shows the structure of the SOAP envelope.

6. Locate the *Text view* icon in the toolbar.

In the following image, the pointer points to the Text view icon.



7. To display the structure of the SOAP envelope as text, click the *Text view* icon.

Retrieving or Updating Web Service Method Connection Information

The <SOAP-ENV:Header> tag is not required and can be deleted from the SOAP envelope.

8. Locate the following section:

```
<m:IMPORTTARGET
xmlns:m="urn:schemas-iwaysoftware-com:dec2002:iwse:af">
<m:targetinstance>
<m:target>String</m:target>
<m:name>String</m:name>
<m:description>String</m:description>
<m:repositoryid>String</m:repositoryid>
<m:exporttime>2001-12-17T09:30:47-05:00</m:exporttime>
<m:contents>R01GODlhcgGSALMAAAQCAEMmCZtuMFQxDS8b</m:contents>
</m:targetinstance>
</m:IMPORTTARGET>
```

- a. For the <m:target> tag, replace the String placeholder with the EIS target system name.
 - b. For the <m:name> tag, replace the String placeholder with the new name of the target you want to import.
 - c. For the <m:description> tag, replace the String placeholder with a description of the target.
 - d. For the <m:repositoryid> tag, copy and paste the contents of the <m:repositoryid> tag that was returned when you exported your target.
 - e. For the <m: exporttime> tag, copy and paste the contents of the <m: exporttime> tag that was returned when you exported your target.
 - f. For the <m: contents> tag, copy and paste the contents of the <m: contents> tag that was returned when you exported your target.
- 9.** From the SOAP menu, select *Send request to server*.

Retrieving or Updating Web Service Method Connection Information

After you migrate your repository, you can retrieve or update connection information for your Web service methods.

Procedure: How to Retrieve Web Service Method Connection Information

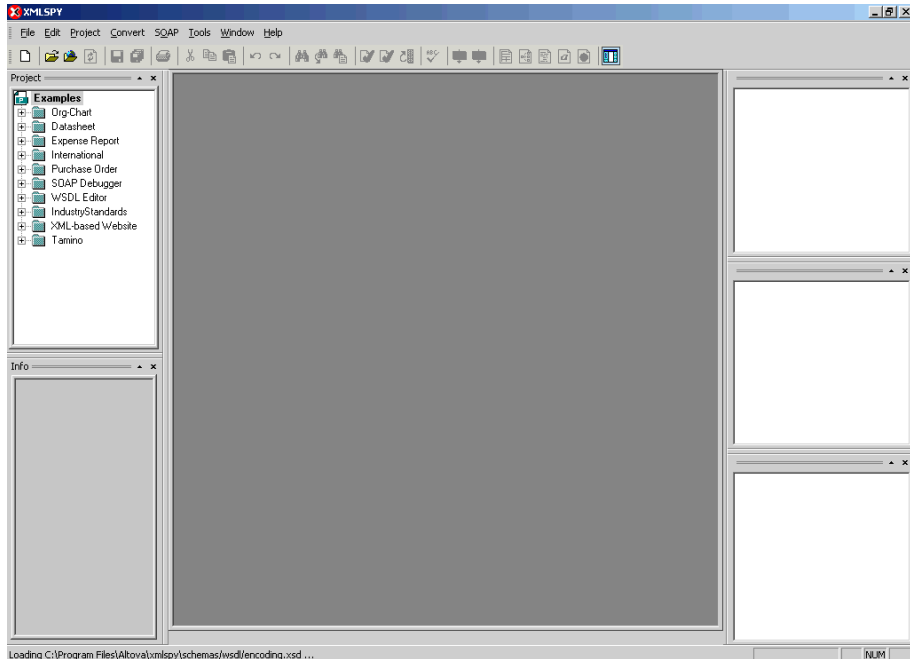
To retrieve Web service method connection information:

1. Copy the iBSE configuration service URL, for example:

```
http://localhost:7777/ibse/IBSEServlet/admin/iwconfig.ibs?wsdl
```

2. Open a third party XML editor, for example, XMLSPY.

The following image shows the XMLSPY window. The upper left has a Project pane that contains a tree of sample files, and the lower left has a blank Info pane. The middle pane is blank. The right side is divided into three blank panes.



3. From the SOAP menu, select *Create new SOAP request*.
The WSDL file location dialog box opens.
4. In the Choose a file field, paste the iBSE configuration service URL, and click *OK*.
The soap operation name dialog box opens and lists the available control methods.
5. Select the *GETMTHCONNECTION(GETMTHCONNECTION parameters)* control method and click *OK*.
A window opens, which shows the structure of the SOAP envelope.
6. Locate the *Text view* icon in the toolbar.
In the following image, the pointer points to the Text view icon.



7. To display the structure of the SOAP envelope as text, click the *Text view* icon.

Retrieving or Updating Web Service Method Connection Information

The <SOAP-ENV:Header> tag is not required and can be deleted from the SOAP envelope.

8. Locate the following section:

```
<m:GETMTHCONNECTION
xmlns:m="urn:schemas-iwaysoftware-com:jul2003:ibse:config">
<m:serviceName>String</m:serviceName>
<m:methodName>String</m:methodName>
</m:GETMTHCONNECTION>
```

- a. For the <m:serviceName> tag, replace the String placeholder with the name of the Web service.
 - b. For the <m:methodName> tag, replace the String placeholder with name of the Web service method.
9. From the SOAP menu, select *Send request to server*.

A response is returned that contains the <m:descriptor> element. You must use this element when updating your Web service method.

Procedure: How to Update Web Service Method Connection Information

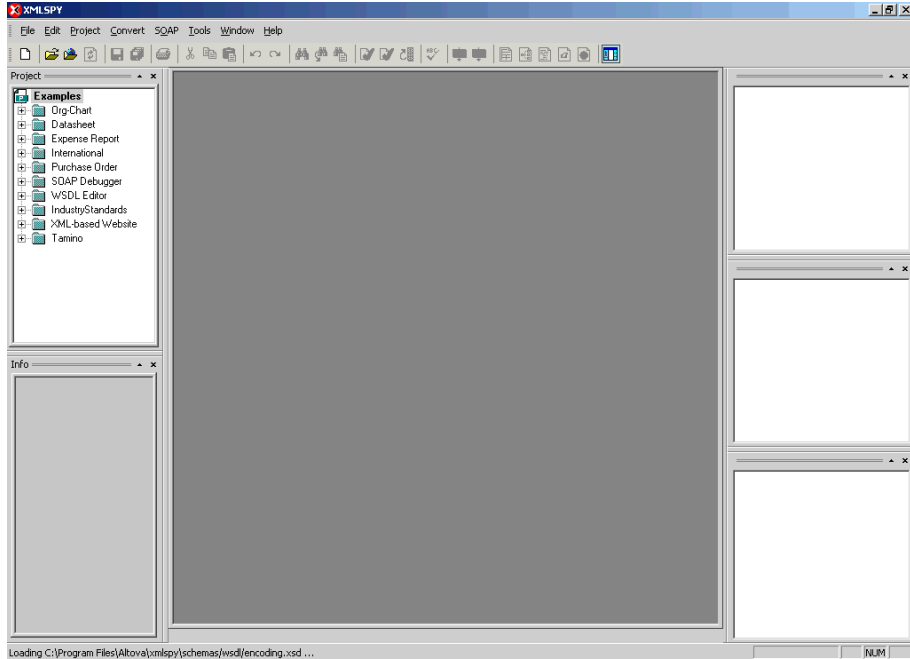
To update Web service method connection information:

1. Copy the iBSE configuration service URL, for example:

```
http://localhost:7777/ibse/IBSEServlet/admin/iwconfig.ibs?wsdl
```

2. Open a third party XML editor, for example, XMLSPY.

The following image shows the XMLSPY window. The upper left has a Project pane that contains a tree of sample files, and the lower left has a blank Info pane. The middle pane is blank. The right side is divided into three blank panes.



3. From the SOAP menu, select *Create new SOAP request*.
The WSDL file location dialog box opens.
4. In the Choose a file field, paste the iBSE configuration service URL, and click *OK*.
The soap operation name dialog box opens and lists the available control methods.
5. Select the *SETMTHCONNECTION(SETMTHCONNECTION parameters)* control method and click *OK*.
A window opens that shows the structure of the SOAP envelope.
6. Locate the *Text view* icon in the toolbar.
In the following image, the pointer points to the Text view icon.



7. To display the structure of the SOAP envelope as text, click the *Text view* icon.

Starting or Stopping a Channel Programmatically

The <SOAP-ENV:Header> tag is not required and can be deleted from the SOAP envelope.

8. Locate the following section:

```
<m:SETMTHCONNECTION
xmlns:m="urn:schemas-iwaysoftware-com:jul2003:ibse:config">
<m:servicename>String</m:servicename>
<m:methodname>String</m:methodname>
<m:descriptor format="" channel="">
  <m:option title="">
    <m:group title="">
      <m:param/>
    </m:group>
  </m:option>
</m:descriptor>
</m:SETMTHCONNECTION>
```

- a. For the <m:servicename> tag, replace the String placeholder with the name of the Web service.
 - b. For the <m:methodname> tag, replace the String placeholder with the name of the Web service method.
 - c. For the <m: descriptor> tag, copy and paste the contents of the <m: descriptor> tag that was returned when you retrieved Web Service method connection information.
9. Modify the contents of the <m: descriptor> tag to change the existing Web Service method connection information.
 10. From the SOAP menu, select *Send request to server*.

Starting or Stopping a Channel Programmatically

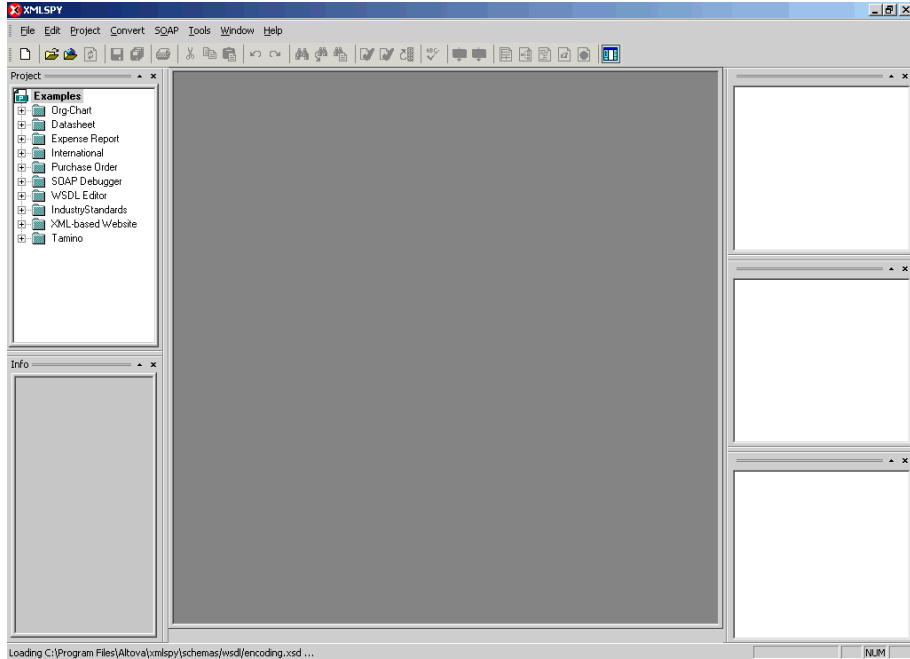
The following topic describes how to start or stop a channel programmatically.

Procedure: How to Start a Channel Programmatically

To start a channel programmatically:

1. Copy the iBSE control event URL, for example:
<http://localhost:7777/ibse/IBSEServlet/admin/iwevent.ibs?wsdl>
2. Open a third party XML editor, for example, XMLSPY.

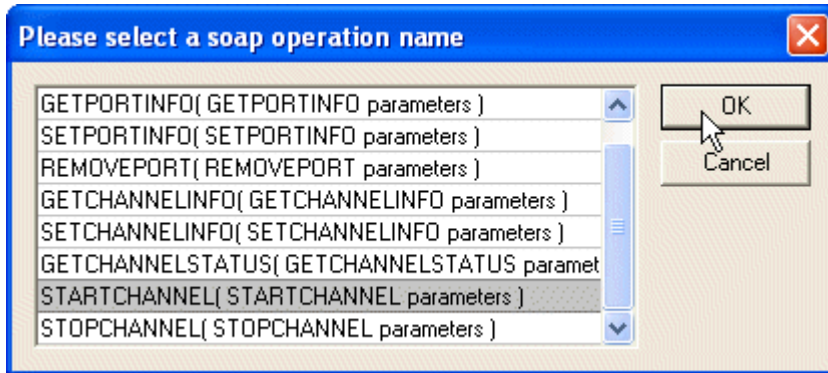
The following image shows the XMLSPY window. The upper left has a Project pane that contains a tree of sample files, and the lower left has a blank Info pane. The middle pane is blank. The right side is divided into three blank panes.



3. From the SOAP menu, select *Create new SOAP request*.
The WSDL file location dialog box opens.
4. In the Choose a file field, paste the iBSE control event URL, and click *OK*.

Starting or Stopping a Channel Programmatically

The following image shows the soap operation name dialog box that opens with a list of available control methods. You can select one and click OK or to escape from the dialog box, you can click Cancel.



5. Select the *STARTCHANNEL(STARTCHANNEL parameters)* control method and click *OK*.

A window opens, which shows the structure of the SOAP envelope.

6. Locate the *Text view* icon in the toolbar.

In the following image, the pointer points to the *Text view* icon.



7. To display the structure of the SOAP envelope as text, click the *Text view* icon.

The `<SOAP-ENV:Header>` tag is not required and can be deleted from the SOAP envelope.

8. Locate the following section:

```
<SOAP-ENV:Body>
  <m:STARTCHANNEL
    xmlns:m="urn:schemas-iwaysoftware-com:dec2002:iwse:event">
    <m:channel>String</m:channel>
  </m:STARTCHANNEL>
</SOAP-ENV:Body>
```

9. For the `<m:channel>` tag, replace the String placeholder with the name of the Channel you want to start.

10. From the SOAP menu, select *Send request to server*.

Procedure: How to Stop a Channel Programmatically

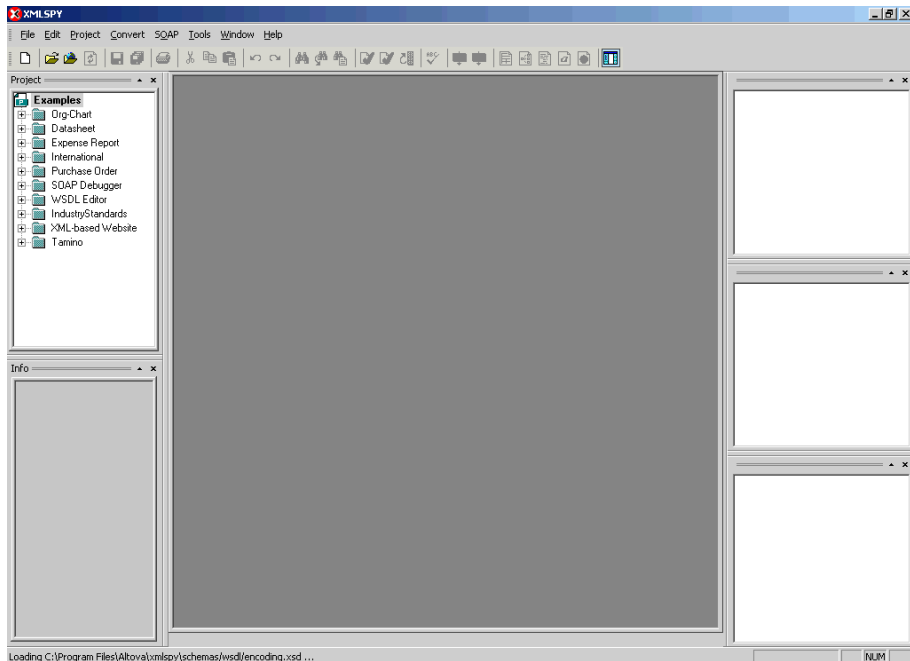
To stop a channel programmatically:

1. Copy the iBSE control event URL, for example:

<http://localhost:7777/ibse/IBSEServlet/admin/iwevent.ibs?wsdl>

2. Open a third party XML editor, for example, XMLSPY.

The following image shows the XMLSPY window. The upper left has a Project pane that contains a tree of sample files, and the lower left has a blank Info pane. The middle pane is blank. The right side is divided into three blank panes.



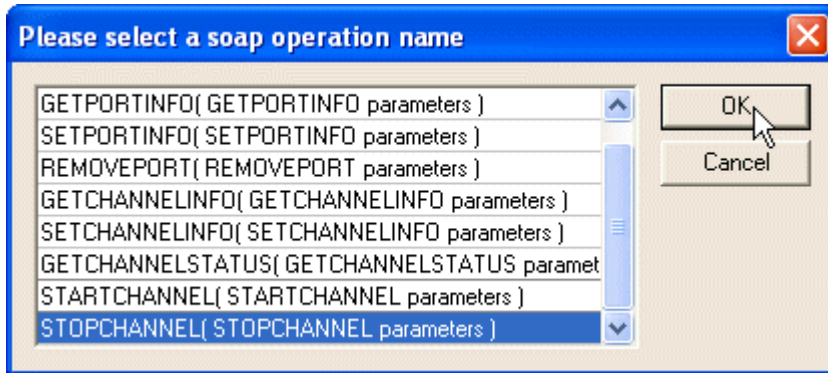
3. From the SOAP menu, select *Create new SOAP request*.

The WSDL file location dialog box opens.

4. In the Choose a file field, paste the iBSE control event URL, and click *OK*.

Starting or Stopping a Channel Programmatically

The following image shows the soap operation name dialog box that opens with a list of available control methods. You can select one and click OK or to escape from the dialog box, you can click Cancel.



5. Select the *STOPCHANNEL(STOPCHANNEL parameters)* control method and click *OK*.

A window opens, which shows the structure of the SOAP envelope.

6. Locate the *Text view* icon in the toolbar.

In the following image, the pointer points to the *Text view* icon.



7. To display the structure of the SOAP envelope as text, click the *Text view* icon.

The `<SOAP-ENV:Header>` tag is not required and can be deleted from the SOAP envelope.

8. Locate the following section:

```
<SOAP-ENV:Body>
  <m:STOPCHANNEL
    xmlns:m="urn:schemas-iwaysoftware-com:dec2002:iwse:event">
    <m:channel>String</m:channel>
  </m:STOPCHANNEL>
</SOAP-ENV:Body>
```

9. For the `<m:channel>` tag, replace the String placeholder with the name of the Channel you want to stop.

10. From the SOAP menu, select *Send request to server*.

APPENDIX A

JDBC Drivers

Topic:

- Copying and Collecting a JDBC File

This section lists the supported JDBC drivers for use with the iWay XML Adapter for RDBMS.

Copying and Collecting a JDBC File

The iWay XML Adapter for RDBMS requires JDBC drivers when connecting to certain databases.

To enable these adapters:

1. Use the following table to determine the JDBC driver files required for your database.
2. Copy the required JDBC files into the iWay 5.5 lib directory.

The default location for this directory on Windows is:

`C:\Program Files\iWay55\lib`

On other platforms, use the corresponding location.

The following table lists the iWay adapters and the required libraries or drivers.

iWay Adapter for...	Required Libraries or Drivers
CICS	None
DB2 Net Driver DB2 App Driver	JDBC driver for DB2 (db2java.zip). Installed as part of the DB2 server. The default location on Windows is one of the following: <code>C:\SQLLIB\java\db2java.zip</code> <code>C:\Program Files\SQLLIB\java\db2java.zip</code> Because the iWay XML Adapter for RDBMS uses JDBC 2.0 features, you must run the usejdbc2.bat file supplied with DB2. Obtain the usejdbc2.bat from your DB2 server or download the current version from the DB2 JDBC driver download Web page. This process builds the proper JDBC 2.0-compliant version of the db2java.zip file. Note: To run the usejdbc2 procedure, you must stop the DB2 instance.
IMS	None

iWay Adapter for...	Required Libraries or Drivers
Informix	<p>JDBC driver for Informix (ifxjdbc.jar).</p> <p>Download the driver from the Informix Web site.</p> <p>For Informix 7: ifxjdbc.jar</p> <p>For Informix 9: ifxjdbc.zip</p>
Oracle E-Business Suite	<p>Oracle JDBC drivers (thin type4 or OCI type2) and/or Oracle Client NET8 or NET9.</p> <p>All calls to Oracle E-Business Suite occur through these drivers. If you do not have the appropriate JDBC driver, Oracle Technology Network (OTN) provides a download site at:</p> <p>http://otn.oracle.com/software/tech/java/sqlj_jdbc/content.html</p> <p>Note: To download the drivers, you require a logon ID.</p> <p>If you are using OCI drivers, you must install and configure Oracle Client on the machine with the iWay Adapter for Oracle E-Business Suite.</p> <p>To use iWay Concurrent Program request functionality, you must install and configure Oracle Client on the Oracle database that supports Oracle E-Business Suite.</p>
Oracle	<p>Oracle JDBC driver (classes12.zip).</p> <p>You can download this driver from the Oracle Web site:</p> <p>http://otn.oracle.com/software/tech/java/sqlj_jdbc/content.html</p> <p>Note: To download the drivers, you require a logon ID.</p> <p>For more information on Oracle JDBC issues, see the Oracle JDBC FAQ:</p> <p>http://otn.oracle.com/tech/java/sqlj_jdbc/htdocs/jjdbc_faq.htm</p>

iWay Adapter for...	Required Libraries or Drivers
SQL Server 2000	<p>SQL Server 2000 JDBC driver.</p> <p>The JDBC driver includes the following three files:</p> <p><code>msbase.jar</code> <code>mssqlserver.jar</code> <code>msutil.jar</code></p> <p>You can download the driver free of charge from: http://microsoft.com</p> <p>To download and run an installation program to install the driver, search the site for SQL Server 2000 JDBC driver. The installation program installs the three driver files in:</p> <p><code>drive:\Program Files\Microsoft SQL Server 2000 Driver for JDBC\lib</code></p> <p>Note: Running the driver installation program is not required. If these files are on another machine, simply copy them.</p>
Sybase	<p>JDBC driver for Sybase servers (jConnect for JDBC).</p> <p>You can download the driver from the Sybase downloads Web site: http://www.sybase.com/downloads</p> <p>Select <i>jConnect for JDBC</i> and review information on the jConnect for JDBC Web site.</p> <p>To extract the JDBC driver, obtain the jConnect.zip that corresponds to your version of Sybase and follow the steps described on the jConnect download page.</p>

Reader Comments

In an ongoing effort to produce effective documentation, the Documentation Services staff at Information Builders welcomes any opinion you can offer regarding this manual.

Please use this form to relay suggestions for improving this publication or to alert us to corrections. Identify specific pages where applicable. You can contact us through the following methods:

Mail: Documentation Services - Customer Support
Information Builders, Inc.
Two Penn Plaza
New York, NY 10121-2898

Fax: (212) 967-0460

E-mail: books_info@ibi.com

Web form: <http://www.informationbuilders.com/bookstore/derf.html>

Name: _____

Company: _____

Address: _____

Telephone: _____ Date: _____

E-mail: _____

Comments:

Reader Comments